

Network Working Group
Request for Comments: 2051
Category: Standards Track

M. Allen
Wall Data Inc.
B. Clouston
Z. Kielczewski
Cisco Systems
W. Kwan
Jupiter Technology Inc.
B. Moore
IBM
October 1996

Definitions of Managed Objects for APPC using SMIV2

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Table of Contents

1.	Introduction	1
2.	The SNMP Network Management Framework	1
3.	Overview	2
3.1	APPC MIB structure	4
4.	Definitions	10
5.	Acknowledgments	123
6.	References	123
7.	Security Considerations	123
8.	Authors' Addresses	124

1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing the configuration, monitoring and controlling of network devices with APPC (Advanced Program-to-Program Communications) capabilities. This memo identifies managed objects for the SNA LU6.2 protocols.

2. The SNMP Network Management Framework

The SNMP Network Management Framework consists of several components. For the purpose of this specification, the applicable components of the Framework are the SMI and related documents [2, 3, 4], which define the mechanisms used for describing and naming objects for the

purpose of management.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

3. Overview

This document identifies the proposed set of objects for managing the configuration, monitoring and controlling devices with APPC capabilities. APPC is the aspect of SNA which supports peer-to-peer communication, and provides the interface for applications to communicate. In this document, we will describe LU6.2 protocol-specific managed objects.

This document describes both dependent and independent LU 6.2 protocols.

A dependent LU requires assistance from an SSCP in order to activate an LU 6.2 session. An independent LU is able to activate an LU 6.2 session without assistance from the SSCP. If the agent supports dependent LU 6.2 only, the SNA NAU MIB, RFC 1666 [7] is used instead to represent those objects.

Local LUs and partner LUs connect with each other using sessions. Multiple different sessions can be established between LUs with characteristics defined by Modes. Session limits within a defined Mode are negotiated between the local and partner LUs using a protocol called CNOS (Change Number of Sessions).

Transaction Programs (TPs) are applications that use sessions to communicate with each other. Multiple TPs can use the same session, but not at the same time. A single usage of a session is called a conversation. While a session can stay active for a long time, a conversation can come up and down based on usage by the TPs.

Common Programming Interface - Communications (CPI-C) is a standard API (Application Programming Interface) for APPC and OSI TP that is used by TPs for accessing conversations. Although, many of the CPI-C objects in this MIB are relevant to both APPC and OSI TP, the intention is for managing APPC products only.

SNA names such as LU names, CP names, mode names, and COS names can be padded with space characters in SNA formats. These space characters are insignificant. For example, in a BIND RU a mode name of "#INTER" with a length of 6 is identical to a mode name of "#INTER " with a length of 8. However, in this MIB, insignificant space characters are not included by the agent. Using the mode name from the previous example, an agent would return a length of 6 and the

string "#INTER" with no space characters for appcModeAdminModeName, regardless of how it appears in the BIND RU or in internal storage. The lone exception is the all blank mode name, for which the agent returns a length of 8 and the string " " (8 space characters).

When an SNA name is functioning as a table index, an agent shall treat trailing space characters as significant. If a Management Station requests the objects from a row with index "#INTER ", the agent does not match this to the row with index "#INTER". Since an agent has no insignificant space characters in any of its table indices, the only reason for a Management Station to include them would be to start GetNext processing at a chosen point in a table. For example, a GetNext request with index "M " would start retrieval from a table at the first row with an 8-character index beginning with M or a letter after M.

The SNA/APPC terms and overall architecture are documented in [1], [5], and [6].

Highlights of the management functions supported by the APPC MIB module include the following:

- o Activating and deactivating statistics keeping and counting.
- o Activating and deactivating tracing.
- o Issuing CNOS processing verbs/commands for INITIALIZE_SESSION_LIMIT, CHANGE_SESSION_LIMIT and RESET_SESSION_LIMIT.
- o Monitoring of parameters related to local LU, partner LU, modes, TPs and CPI-C side information.
- o Deactivating sessions.
- o Monitoring of LU6.2-specific session operational parameters and statistics, historical information about abnormally terminated sessions, and information about APPC sessions that are transported by APPN HPR.
- o Monitoring of conversation operational parameters, and historical information about abnormally terminated sessions.

This MIB module does not support:

- o Modifying APPC defaults.
- o Creating and deleting partner LUs, modes, TPs, and CPI-C side information tables.
- o Modifying parameters related to local LU, partner LU, modes, TPs, and CPI-C side information.
- o Activating or deactivating local LUs.
- o Activating or deactivating partner LUs.
- o Activating or deactivating conversations.
- o Activating or deactivating Transaction Programs.
- o Activating sessions.
- o Traps

3.1. APPC MIB Structure

The APPC MIB module contains six groups of objects:

- o appcGlobal - objects related to global defaults and controls.
In addition, CNOS processing objects are also part of this group.
- o appcLu - objects related to LU6.2-specific local and partner LU, mode definition, monitoring and control.
- o appcTp - objects related to transaction program definition, monitoring and control.
- o appcSession - objects related to LU6.2-specific session monitoring.
- o appcConversation - objects related to conversation monitoring.
- o appcCPIC - objects related to related CPI-C side information.

These groups are described below in more detail.

The objects related to LU6.2 are generally organized into two types of tables: the Admin and Oper tables.

The "Admin" table contains read-only objects which contain default or expected configuration values. This MIB does not create or modify configuration values. The "Oper" table contains objects which provide current operational values, such as state values or negotiated parameters, for dynamic or configured objects. Dynamic objects are created by the APPC system using one of the templates provided in the "Admin" table. Configured objects usually have a one-to-one relationship between "Admin" and "Oper" entries. However, some "Admin" values may have changed since the object became operational, such that the "Oper" values may no longer be based on the "Admin" values. The "Admin" entry could even be deleted. For example, some implementations may allow a mode definition (appcModeAdminEntry) to be deleted even while an active session was using this mode (appcModeOperEntry still exists). Where appropriate, the "Oper" table may include initial starting values for objects that can be reconfigured while operational. How the "Admin" values are changed or deleted is outside the scope of this MIB.

3.1.1. appcGlobal group

The appcGlobal group consists of the following tables and objects:

1) appcCntrlAdminGroup

This group of objects controls whether certain statistics and counters (e.g., session counters and RSCV collection) should be maintained by the Agent. In addition, the ability to activate and deactivate tracing is also supported through objects in this group. These objects are for Agent implementations that wish to provide this level of operational control and are optional.

The objects in this group represent the desired state, with the actual operational values in appcCntrlOperGroup.

These objects can be generated initially, after startup of SNA service, by the Agent which uses information from the Node configuration file. Subsequent modifications of object values is possible by a Management station. The modifications to these objects can be saved in the Node configuration file for the next startup (i.e., restart or next initialization) of SNA service, but the mechanism for this function is not defined in this document.

2) appcCntrlOperGroup

This group of objects monitors whether certain statistics and counters (e.g., session counters and RSCV collection) are maintained by an Agent. In addition, the ability to monitor tracing activity is also supported through objects in this group.

This table represents the actual operational state. These states can be modified via objects in the appcCntrlAdminGroup.

3) appcGlobalObjects

These objects describe global information such as APPC system start time, the control point name, and default LU 6.2 configuration values. The type of default configuration information includes mode name, LU, and maximum logical record size.

4) appcCnosControl

These objects allows for issuing of CNOS commands relative to a local and partner LU pair and a Mode. They support the following CNOS commands: INITIALIZE_SESSION_LIMIT, CHANGE_SESSION_LIMIT and RESET_SESSION_LIMIT.

The objects in this group can be modified by a Management Station.

This group consists of objects that are relevant to the CNOS commands parameters, which a Management Station needs to set. After setting the parameters of a CNOS command, the Management Station will set the control object (appcCnosCommand) to request the Agent to issue the appropriate CNOS command.

3.1.2. appcLu group

The appcLu group consists of the following tables:

1) appcLluAdminTable

This table contains objects which describe specific LU6.2 local LU configuration information. The type of information includes the maximum number of sessions supported and compression parameters.

2) appcLluOperTable

This table contains objects which describe specific LU6.2 local LU operational information. The type of information includes the maximum number of sessions supported, the number of sessions currently active, and compression parameters.

3) appcLuPairAdminTable

This table contains objects which describe local LU and partner LU configuration information. The type of information includes security information and whether parallel sessions are supported.

For those implementations that have partner LU definitions associated with each local LU, multiple entries with the same appcLuPairAdminParLuName could exist with different appcLuPairAdminLocLuName. For those implementations in which partner LU definitions apply to all local LUs, the appcLuPairAdminLocLuName is set to '*ALL'.

4) appcLuPairOperTable

This table contains objects which describe partner/local LU pair run-time operational information. The type of information includes security information and whether parallel sessions are supported.

Although the Admin (appcLuPairAdminTable) table entries could be global to all local LUs in a Node, an entry in this Oper table is always associated with one local LU.

A row in this table is created as soon as there is an active session between the local and partner LU. Two entries are present when both LUs in a pair are local.

5) appcModeAdminTable

This table contains objects which describe Mode configuration information. The type of information includes the mode name and maximum session limit.

For those implementations that have Mode definitions associated with each local and partner LU pair, multiple entries with the same appcModeAdminModeName could exist with different appcModeAdminLocLuName and appcModeAdminParLuName. For those implementations in which Mode definitions apply to all local and/or all partner LUs, the appcModeAdminLocLuName and/or appcModeAdminParLuName are set to '*ALL'.

6) appcModeOperTable

This table contains objects which describe Mode run-time operational information for each local/partner LU pair. The type of information includes the mode name and maximum session limit.

Although the Admin table entries could be global to all local and partner LUs in a Node, the Oper table entries are always associated with one local and partner LU pair.

A row in this table is created as soon as there is an active session between local and partner LU for this Mode. Two entries are present when both LUs in a pair are local.

3.1.3. appcTp group

The appcTp group consists of the following table:

1) appcTpAdminTable

This table contains objects which describe transaction program (TP) configuration information. The type of information includes the TP name and TP operation, indicating how the TP will be started.

For those implementations that have TP definitions associated with each local LU, multiple entries with the same appcTpAdminTpName could exist with different appcTpAdminLocLuName. For those implementations in which TP definition applies to all local LUs, it will have appcTpAdminLocLuName set to '*ALL'.

There is no appcTpOperTable. Run-time information about TP tends to be product-specific (e.g., process Id), and much of the information can be derived from the conversation tables.

3.1.4. appcSession group

The appcSession group consists of the following tables:

1) appcActSessTable

This table contains objects which describe LU6.2 session information. The type of information includes the PCID and the pacing counts.

2) appcSessStatsTable

This table contains statistical information about LU 6.2 sessions. The type of information includes counters of bytes and RUs sent and received.

3) appcHistSessTable

This table contains historical information about APPC sessions that have terminated abnormally. The type of information includes the unbind type and sense data.

4) appcSessRtpTable

This table contains information about LU 6.2 sessions that are being transported by High Performance Routing. The type of information includes the NCEID and TCID.

3.1.5. appcConversation group

The appcConversation group consists of the following tables:

1) appcActiveConvTable

This table contains objects which describe active conversation information. The type of information includes the state and type. An entry is created by an Agent when the conversation is started, and is removed when the conversation ends.

2) appcHistConvTable

This table contains objects which describe historical conversation information about abnormally terminated conversations. The number of entries and how long they are kept depends on the Agent implementation. The type of information includes the sense data and log data.

3.1.6. appcCPIC group

The appcCPIC group consists of the following tables:

1) appcCpicAdminTable

This table contains objects which describe CPI-C side information. The type of information includes the symbolic destination name and partner LU name.

For those implementations that have CPI-C definition associated with each local LU, multiple entries with the same appcCpicAdminSymbDestName could exist with different appcCpicAdminLocLuName. For those implementations in which CPI-C definition applies to all local LUs, it will have appcCpicAdminLocLuName set to '*ALL'.

2) appcCpicOperTable

This table contains objects which describe CPI-C run-time operational information. The type of information includes the symbolic destination name and partner LU name.

4. Definitions

```
APPC-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    DisplayString, InstancePointer, TEXTUAL-CONVENTION, DateAndTime
    FROM SNMPv2-TC
```

```
    mib-2, Counter32, Gauge32, Integer32, TimeTicks,
    OBJECT-TYPE, MODULE-IDENTITY
    FROM SNMPv2-SMI
```

```
    snanauMIB
    FROM SNA-NAU-MIB
```

```
    MODULE-COMPLIANCE, OBJECT-GROUP
    FROM SNMPv2-CONF;
```

```
appcMIB MODULE-IDENTITY
```

```
    LAST-UPDATED   "9512150000Z"
    ORGANIZATION   "IETF SNA NAU MIB Working Group"
    CONTACT-INFO
```

```
    "
```

```
        Michael Allen
        Wall Data Inc.
        P.O.Box 1120
        Duval, WA 98019, USA
        Tel:      1 206 844 3505
        E-mail:   mallen@hq.walldata.com
```

```
        Bob Clouston
        Cisco Systems
        7025 Kit Creek Road
        P.O. Box 14987
        Research Triangle Park, NC 27709, USA
        Tel:      1 919 472 2333
        E-mail:   clouston@cisco.com
```

```
        Zbigniew Kielczewski
        Cisco Systems
        3100 Smoketree Court
        Raleigh, NC 27604, USA
        Tel:      1 919 871 6326
        E-mail:   zbig@cisco.com
```

William Kwan
 Jupiter Technology Inc.
 200 Prospect Street
 Waltham, MA 02254, USA
 Tel: 1 617 894 9300, x423
 E-mail: billk@jti.com

Bob Moore
 IBM Corporation
 800 Park Offices Drive
 CNMA/664
 P.O. Box 12195
 Research Triangle Park, NC 27709, USA
 Tel: 1 919 254 4436
 E-mail: remoore@ralvm6.vnet.ibm.com

"

DESCRIPTION

"This is the MIB module for objects used to manage network devices with APPC capabilities."

```
::= { snanauMIB 3 }
```

```
appcObjects      OBJECT IDENTIFIER ::= { appcMIB 1 }
  appcGlobal      OBJECT IDENTIFIER ::= { appcObjects 1 }
  appcLu          OBJECT IDENTIFIER ::= { appcObjects 2 }
  appcTp          OBJECT IDENTIFIER ::= { appcObjects 3 }
  appcSession     OBJECT IDENTIFIER ::= { appcObjects 4 }
  appcConversation OBJECT IDENTIFIER ::= { appcObjects 5 }
  appcCPIC        OBJECT IDENTIFIER ::= { appcObjects 6 }
```

```
-- *****
-- Objects in this MIB are used to model an SNA device that supports
-- APPC LUs.
-- Following is the overall organization of the MIB.
--
-- 1.    APPC Global Objects          - global values, defaults,
--                                   - controls (including CNOS)
-- 2.    APPC Defined Lu Tables       - Admin and Oper
-- 3.    APPC Defined LU Pair Tables  - Admin and Oper
-- 4.    APPC Mode Tables             - Admin and Oper
-- 5.    APPC TP Tables               - Admin only
-- 6.    APPC Session Tables          - Active, Stats, History, RTP
-- 7.    APPC Conversation Table      - Active, History
-- 8.    APPC CPIC side info          - Admin and Oper
-- *****
-- *****
```

-- Textual Convention

SnaSenseData ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"To facilitate their display by a Management Station, sense data objects in the MIB are represented as DisplayStrings of size 8. Eight '0' characters indicates that no sense data identifying an SNA error condition is available."

SYNTAX DisplayString (SIZE (8))

-- *****

-- APPC Control Objects

-- The following objects allow:

-- * the collection of APPC Session information counters
-- to be started and stopped

-- * the collection of APPC Session RSCVs
-- to be started and stopped

-- * the collection of APPC tracing information to be started and
-- stopped

--

-- These objects are for implementations that wish to provide

-- this level of operational control. This group is

-- conditionally mandatory in the conformance section of the MIB.

--

-- *****

-- *****

-- Control Admin

-- These objects contain the desired states for the controls.

-- The actual states are in the Oper objects.

-- *****

appcCntrlAdminGroup OBJECT IDENTIFIER ::= { appcGlobal 1 }

appcCntrlAdminStat OBJECT-TYPE

SYNTAX INTEGER {
notActive(1),
active(2)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Indicates the desired state of statistics collection:

notActive collection of counters is not active.

active collection of counters is active.

When this object is set to notActive, all of the entries are removed from the appcSessStatsTable."

```
::= { appcCntrlAdminGroup 1 }
```

```
appcCntrlAdminRscv OBJECT-TYPE
```

```
SYNTAX INTEGER {
    notActive(1),
    active(2)
}
```

```
MAX-ACCESS read-write
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Indicates the desired state of RSCV information collection:
    notActive  collection of route selection control vectors
                is not active.
    active     collection of route selection control vectors
                is active."
```

```
::= { appcCntrlAdminGroup 2 }
```

```
appcCntrlAdminTrace OBJECT-TYPE
```

```
SYNTAX INTEGER {
    notActive(1),
    active(2)
}
```

```
MAX-ACCESS read-write
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Indicates the desired state of tracing:
```

```
    notActive  collection of tracing information is not active
    active     collection of tracing information is active"
```

```
::= { appcCntrlAdminGroup 3 }
```

```
appcCntrlAdminTraceParm OBJECT-TYPE
```

```
SYNTAX DisplayString (SIZE (0..128))
```

```
MAX-ACCESS read-write
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Specifies the parameter to be used in conjunction with
activating tracing. The actual content is implementation
dependent."
```

```
::= { appcCntrlAdminGroup 4 }
```

```
-- *****
```

```
-- Control Oper
--      These objects contain the actual states of the controls.
-- *****
appcCntrlOperGroup OBJECT IDENTIFIER ::= { appcGlobal 2 }
```

appcCntrlOperStat OBJECT-TYPE

```
    SYNTAX INTEGER {
        notActive(1),
        active(2)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Indicates the current collection options in effect:

        notActive  collection of counters is not active.
        active     collection of counters is active.

        Statistical entries are present in the appcSessStatsTable
        only when the value of this object is 'active'."

    ::= { appcCntrlOperGroup 1 }
```

appcCntrlOperStatTime OBJECT-TYPE

```
    SYNTAX TimeTicks
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Time since the appcCntrlOperStat object last changed.
        This time is in hundreds of a second."

    ::= { appcCntrlOperGroup 2 }
```

appcCntrlOperRscv OBJECT-TYPE

```
    SYNTAX INTEGER {
        notActive(1),
        active(2)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Indicates the current collection options in effect:

        notActive  collection of route selection control vectors
                    is not active.
        active     collection of route selection control vectors
                    is active."
```

```
::= { appcCntrlOperGroup 3 }
```

```
appcCntrlOperRscvTime OBJECT-TYPE
```

```
SYNTAX TimeTicks
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Time since the appcCntrlOperRscv object last changed.
   This time is in hundreds of a second."
```

```
::= { appcCntrlOperGroup 4 }
```

```
appcCntrlOperTrace OBJECT-TYPE
```

```
SYNTAX INTEGER {
    notActive(1),
    active(2)
}
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Indicates the current state of tracing:
```

```
    notActive  collection of tracing information is not active.
    active     collection of tracing information is active."
```

```
::= { appcCntrlOperGroup 5 }
```

```
appcCntrlOperTraceTime OBJECT-TYPE
```

```
SYNTAX TimeTicks
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Time since the appcCntrlOperTrace object last changed.
   This time is in hundreds of a second."
```

```
::= { appcCntrlOperGroup 6 }
```

```
appcCntrlOperTraceParm OBJECT-TYPE
```

```
SYNTAX DisplayString (SIZE (0..128))
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Specifies the parameter used in conjunction with activating
   tracing. The actual content is implementation dependent."
```

```
::= { appcCntrlOperGroup 7 }
```

```
-- *****
```

```
--
--   APPC global settings
--
-- *****
appcGlobalObjects OBJECT IDENTIFIER ::= { appcGlobal 3 }

appcUpTime OBJECT-TYPE
    SYNTAX TimeTicks
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The time, in hundredths of a second, since the
        APPC portion of the system was last reinitialized."

    ::= { appcGlobalObjects 1 }

appcDefaultModeName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..8))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Specifies the mode name to be used under the following
        conditions:

        When an incoming BIND request contains a mode name not
        defined at the local node. The parameters defined for
        this mode are used for the inbound implicit mode
        capability.

        When an APPC program issues an [MC_]ALLOCATE,
        [MC_]SEND_CONVERSATION, or CNOS verb, or when a CPI-C
        program issues an Allocate (CMALLC) call,
        specifying a mode name not defined at the local node. The
        parameters defined for this mode are used for the outbound
        implicit mode capability.

        This mode name must match a defined entry in the
        appcModeAdminTable."

    ::= { appcGlobalObjects 2 }

appcDefaultLuName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..17))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Specifies the name of the local LU that is to serve as the
        default LU. This is the default LU to which are routed inbound
```


BIND requests that exclude the secondary LU name. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present. This local LU name must match a defined entry in the appcLluAdminTable."

::= { appcGlobalObjects 3 }

appcDefaultImplInbndPlu OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether or not inbound implicit partner LU support is enabled. The following values are defined:

- no - Specifies that inbound implicit partner LU support is disabled, which means that an incoming bind that specifies a partner LU that is not defined at the local node will be rejected.
- yes - Specifies that inbound implicit partner LU support is enabled, which provides the capability to accept an incoming BIND request that contains a partner LU name that is not defined at the local node."

::= { appcGlobalObjects 4 }

appcDefaultMaxMcLlSndSize OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the maximum size of a logical record to be used for a mapped conversation when sending data to either the inbound or outbound implicit partner LU. This size is the maximum number of bytes in a single logical record, as indicated in the LL field of the record. The default value is 32767.

Note that this object does not limit the maximum size that an application program can supply on the Send Data call for a mapped conversation."

::= { appcGlobalObjects 5 }

appcDefaultFileSpec OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..80))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The local file specification that is to be searched by the APPC attach manager when no DEFINE_TP verb has been issued for the TP name received on an incoming attach. In this case, the attach manager will attempt to start a program whose file name is the same as the incoming TP name. If found, the program is loaded. If not found, the attach is rejected.

The value '*' indicates that the normal search path for executable programs is to be used for locating an undefined transaction program.

A null string indicates that there is no default file specification for undefined transaction programs."

```
::= { appcGlobalObjects 6 }
```

appcDefaultTpOperation OBJECT-TYPE

SYNTAX INTEGER {

other(1),

queuedOperatorStarted(2),

queuedOperatorPreloaded(3),

queuedAmStarted(4),

nonqueuedAmStarted(5)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies how the program will be started.

other - Specifies that the default TP operation is none of the methods specified below. It may be a product-specific method.

queuedOperatorStarted - Specifies that one version of the program will be run at a time. If an incoming attach arrives and the program has not been started yet, APPC will issue a message to the operator to start the specified program. Subsequent attaches that arrive while the program is active will be queued.

queuedOperatorPreloaded - Specifies that one version

of the program will be run at a time. If an incoming attach arrives and the program has not been started yet, the Attach will be rejected. The APPC attach manager determines that a TP has started upon reception of an APPC RECEIVE_ALLOCATE verb, or a CPI-C Accept_Conversation (CMACCP) or Specify_Local_TP_Name (CMSLTP) call. No message is sent to the operator. Subsequent attaches that arrive while the program is active are queued.

queuedAmStarted - Specifies that one version of the program will be run at a time and will be started by the APPC attach manager. Subsequent attaches that arrive while the program is active will be queued.

nonqueuedAmStarted - Specifies that multiple copies of the program will be run at a time and will be started by the APPC attach manager. "

::= { appcGlobalObjects 7 }

appcDefaultTpConvSecRqd OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether or not conversation security will be used for default TPs.

- no - Specifies that the incoming attach does not have to contain security information.
- yes - Specifies that the incoming attach must contain valid authentication information (e.g., user ID and password)."

::= { appcGlobalObjects 8 }

appcLocalCpName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..17))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the name of the local control point. This field is from 0 to 17 characters in length, including a period (.) which

separates the NetId from the NAU name if the NetId is present.
A null string indicates that the value is unknown."

::= { appcGlobalObjects 9 }

appcActiveSessions OBJECT-TYPE

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the total number of active APPC sessions supported by this implementation. Sessions for which both LUs are local are counted twice."

::= { appcGlobalObjects 10 }

appcActiveHprSessions OBJECT-TYPE

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the total number of active HPR APPC sessions."

::= { appcGlobalObjects 11 }

```
-- *****
-- APPC CNOS control
--
-- This group contains objects for issuing APPC Change-Number-of-Session
-- (CNOS) commands to a specific mode. Specifically, the commands
-- supported are:
--             INITIALIZE_SESSION_LIMIT
--             CHANGE_SESSION_LIMIT
--             RESET_SESSION_LIMIT
--
-- *****
appcCnosControl OBJECT IDENTIFIER ::= { appcGlobal 4 }
```

appcCnosCommand OBJECT-TYPE

SYNTAX INTEGER {
initSesslimit(1),
changeSesslimit(2),
resetSesslimit(3)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Specifies the CNOS command or verb to issue. First set the values of the particular CNOS parameter objects (from the range { appcCnosControl 2 } through { appcCnosControl 8 }) that apply to the CNOS command to be executed, set the three CNOS target objects ({ appcCnosControl 9 } through { appcCnosControl 11 })), then set this object to the command to be executed.

Here is the list of parameter objects that must be set for each of the CNOS commands:

```
INIT_SESSION_LIMIT -
    appcCnosMaxSessLimit
    appcCnosMinCwinLimit
    appcCnosMinClosLimit
    appcCnosTargetLocLuName
    appcCnosTargetParLuName
    appcCnosTargetModeName
```

```
CHANGE_SESSION_LIMIT -
    appcCnosMaxSessLimit
    appcCnosMinCwinLimit
    appcCnosMinClosLimit
    appcCnosResponsible
    appcCnosTargetLocLuName
    appcCnosTargetParLuName
    appcCnosTargetModeName
```

```
RESET_SESSION_LIMIT -
    appcCnosResponsible
    appcCnosDrainPart
    appcCnosForce
    appcCnosTargetLocLuName
    appcCnosTargetParLuName
    appcCnosTargetModeName
```

"

```
::= { appcCnosControl 1 }
```

appcCnosMaxSessLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Specifies the maximum value that the local LU is to use, during CNOS processing, for the session limit. The local LU, as a target LU, will negotiate a higher session limit it receives in the CNOS request down to this maximum value. The

local LU, as a source LU, will restrict the session limit it sends in a CNOS request to a value less than or equal to this maximum value.

If set (i.e., greater than 0), this overrides the maximum session limit defined in the appcModeAdminTable.

This parameter should be set to the desired value before setting the command (appcCnosCommand).

This parameter applies to the INITIALIZE_SESSION_LIMIT and CHANGE_SESSION_LIMIT verbs."

DEFVAL { 0 }

::= { appcCnosControl 2 }

appcCnosMinCwinLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Specifies the default minimum contention winner sessions limit.

This parameter should be set to the desired value before setting the command (appcCnosCommand).

This parameter applies to the INITIALIZE_SESSION_LIMIT and CHANGE_SESSION_LIMIT verbs."

DEFVAL { 0 }

::= { appcCnosControl 3 }

appcCnosMinClosLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Specifies the default minimum contention loser sessions limit.

This parameter should be set to the desired value before setting the command (appcCnosCommand).

This parameter applies to the INITIALIZE_SESSION_LIMIT and CHANGE_SESSION_LIMIT verbs."

DEFVAL { 0 }

::= { appcCnosControl 4 }

appcCnosDrainSelf OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Specifies whether the local LU is draining its conversations for this mode. When a mode session limit is reset (via a CNOS RESET_SESSION_LIMIT request), the local LU could be set to process all queued conversations before deactivating all of the sessions (using the SNA Bracket Initiation Stopped or BIS protocol).

This parameter should be set to the desired value before setting the command (appcCnosCommand).

This parameter applies only to the RESET_SESSION_LIMIT verb."

DEFVAL { no }

::= { appcCnosControl 5 }

appcCnosDrainPart OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Specifies whether the partner LU is draining its conversations for this mode. When a mode session limit is reset (via a CNOS RESET_SESSION_LIMIT request), the partner LU could be set to process all queued conversations before deactivating all of the sessions (using the SNA Bracket Initiation Stop or BIS protocol).

This parameter should be set to the desired value before setting the command (appcCnosCommand).

This parameter applies only to the RESET_SESSION_LIMIT verb."

DEFVAL { yes }

::= { appcCnosControl 6 }

appcCnosResponsible OBJECT-TYPE

SYNTAX INTEGER {
 source(1),
 target(2)
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Specifies which LU is responsible for selecting and deactivating sessions as a result of a change that decreases the session limit or the maximum number of contention winner sessions for the source or target LU. If no session need to be deactivated, this parameter is ignored.

source	-	specifies that the source (local) LU is responsible. The target (partner) LU cannot negotiate this value.
target	-	specifies that the target (partner) LU is responsible. The target LU can negotiate this value to source.

This parameter should be set to the desired value before setting the command (appcCnosCommand).

This parameter applies to the RESET_SESSION_LIMIT and CHANGE_SESSION_LIMIT verbs."

DEFVAL { source }

::= { appcCnosControl 7 }

appcCnosForce OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Specifies whether the local LU should force the resetting of the session limit when certain error conditions occur that prevent the successful exchange of CNOS request and reply.

This parameter should be set to the desired value before

setting the command (appcCnosCommand).

This parameter applies only to the RESET_SESSION_LIMIT verb."

DEFVAL { no }

::= { appcCnosControl 8 }

appcCnosTargetLocLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The SNA name of the local LU to which the CNOS command is to be applied. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

This object should be set to the desired value before setting the command (appcCnosCommand).

This parameter applies to all CNOS verbs."

::= { appcCnosControl 9 }

appcCnosTargetParLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The SNA name of the partner LU to which the CNOS command is to be applied. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

This object should be set to the desired value before setting the command (appcCnosCommand).

This parameter applies to all CNOS verbs."

::= { appcCnosControl 10 }

appcCnosTargetModeName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..8))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Specifies the mode name to which the CNOS command is to be

applied.

This object should be set to the desired value before setting the command (appcCnosCommand).

This parameter applies to all CNOS verbs."

```
::= { appcCnosControl 11 }
```

```
-- *****
--   APPC LU information
--   -----
--   Local LU
--   Partner LU
--   Mode
-- *****
-- *****
--   APPC Local LU
--
--   The entries in the following tables provide information for
--   independent and dependent LU 6.2.
--
-- *****
-- *****
--   APPC Local LU Admin Table
--   Objects in this table contain default or expected configuration
--   values for local 6.2 LUs.
-- *****
```

```
appcLluAdminTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AppcLluAdminEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "APPC Local LU Admin Table."
```

```
::= { appcLu 1 }
```

```
appcLluAdminEntry OBJECT-TYPE
    SYNTAX AppcLluAdminEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Information about local APPC LUs. "
```

INDEX { appcLluAdminName }

::= { appcLluAdminTable 1 }

```
AppcLluAdminEntry ::= SEQUENCE {
    appcLluAdminName          DisplayString,
    appcLluAdminDepType       INTEGER,
    appcLluAdminLocalAddress  OCTET STRING,
    appcLluAdminSessLimit     Integer32,
    appcLluAdminBindRspMayQ   INTEGER,
    appcLluAdminCompression  INTEGER,
    appcLluAdminInBoundCompLevel  INTEGER,
    appcLluAdminOutBoundCompLevel  INTEGER,
    appcLluAdminCompRleBeforeLZ  INTEGER,
    appcLluAdminAlias         DisplayString
}
```

appcLluAdminName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Specifies the name of the local LU. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present."

::= { appcLluAdminEntry 1 }

appcLluAdminDepType OBJECT-TYPE

```
SYNTAX INTEGER {
    dependent(1),
    independent(2)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This value identifies whether the LU is dependent or independent."

::= { appcLluAdminEntry 2 }

appcLluAdminLocalAddress OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (1))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The local address for this LU is a byte with a value ranging from 0 to 254. For dependent LUs, this value ranges from 1 to

254; for independent LUs this value is always 0."

::= { appcLluAdminEntry 3 }

appcLluAdminSessLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum number of sessions supported by this LU."

::= { appcLluAdminEntry 4 }

appcLluAdminBindRspMayQ OBJECT-TYPE

SYNTAX INTEGER {
no(1),
yes(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates whether or not the local LU, as the sender of a BIND request, allows a partner partner LU to delay sending the BIND response if the partner LU cannot process the BIND request immediately."

::= { appcLluAdminEntry 5 }

appcLluAdminCompression OBJECT-TYPE

SYNTAX INTEGER {
prohibited(1),
required(2),
negotiable(3)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether compression is supported. The local LU uses this value for negotiation during session activation (SNA BIND).

prohibited - specifies that no compression is to be used.
required - specifies that compression is required.
negotiable - specifies that the usage of compression is to be negotiated between the LUs. The level of compression is also negotiated."

::= { appcLluAdminEntry 6 }

appcLluAdminInBoundCompLevel OBJECT-TYPE

```

SYNTAX INTEGER {
    none(1),
    rle(2),
    lz9(3),
    lz10(4),
    lz12(5)
}

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the maximum level of compression supported for inbound data. The local LU uses this value in conjunction with appcLluAdminCompression for negotiation during session activation (SNA BIND).

- none - specifies that no compression is to be used.
- rle - specifies run-length encoding compression in which a 1 or 2 byte sequence substitution is used for each repeated run of the same character.
- lz9 - specifies Lempel-Ziv-like compression in which 9 bit codes are used to substitute repeated substrings in the data stream. These codes are indices that refer to entries in a common dictionary generated adaptively at both sender and receiver as the data flows and compression occurs. The larger number bits used for the code, the more storage space is required for the dictionary, but the larger the compression ratio.
- lz10 - specifies a 10 bit code Lempel-Ziv-like compression.
- lz12 - specifies a 12 bit code Lempel-Ziv-like compression."

```

 ::= { appcLluAdminEntry 7 }

```

appcLluAdminOutBoundCompLevel OBJECT-TYPE

```

SYNTAX INTEGER {
    none(1),
    rle(2),
    lz9(3),
    lz10(4),
    lz12(5)
}

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the maximum level of compression supported for outbound data. The local LU uses this value in conjunction with appcLluAdminCompression for negotiation during session activation (SNA BIND).

none - specifies that no compression is to be used.
 rle - specifies run-length encoding compression
 in which a 1 or 2 byte sequence substitution is
 used for each repeated run of the same character.
 lz9 - specifies Lempel-Ziv-like compression in which
 9 bit codes are used to substitute repeated
 substrings in the data stream. These codes are
 indices that refer to entries in a common
 dictionary generated adaptively at both sender and
 receiver as the data flows and compression occurs.
 The larger of number bits used for the code, the
 more storage space is required for the dictionary,
 but the larger the compression ratio.
 lz10 - specifies a 10 bit code Lempel-Ziv-like compression.
 lz12 - specifies a 12 bit code Lempel-Ziv-like compression."

::= { appcLluAdminEntry 8 }

appcLluAdminCompRleBeforeLZ OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
 }
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"Specifies whether run-length encoding is to be applied to the
 data before applying Lempel-Ziv-like compression. The local LU
 uses this value for negotiation during session activation (SNA
 BIND). This parameter is only supported if LZ compression is
 used."

::= { appcLluAdminEntry 9 }

appcLluAdminAlias OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..8))
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"A local alias for the local LU. If not known or
 not applicable, this object contains a zero-length
 string."

::= { appcLluAdminEntry 10 }

-- *****
 -- APPC Local LU Oper Table

```
--      Objects in this table contain current operational values, such
--      as state values or negotiated parameters, for local 6.2 LUs.
--      *****

appcLluOperTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AppcLluOperEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "APPC Local LU Operational Table."
    ::= { appcLu 2 }

appcLluOperEntry OBJECT-TYPE
    SYNTAX AppcLluOperEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Information about local APPC LUs."

    INDEX { appcLluOperName }

    ::= { appcLluOperTable 1 }

AppcLluOperEntry ::= SEQUENCE {
    appcLluOperName          DisplayString,
    appcLluOperDepType       INTEGER,
    appcLluOperLocalAddress  OCTET STRING,
    appcLluOperSessLimit     Integer32,
    appcLluOperBindRspMayQ   INTEGER,
    appcLluOperCompression  INTEGER,
    appcLluOperInBoundCompLevel  INTEGER,
    appcLluOperOutBoundCompLevel  INTEGER,
    appcLluOperCompRleBeforeLZ  INTEGER,
    appcLluOperAlias         DisplayString,
    appcLluOperActiveSessions Gauge32
}

appcLluOperName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..17))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Specifies the name of the local LU.  This field is from 1 to
        17 characters in length, including a period (.) which separates
        the NetId from the NAU name if the NetId is present."

    ::= { appcLluOperEntry 1 }
```

appcLluOperDepType OBJECT-TYPE

SYNTAX INTEGER {
 dependent(1),
 independent(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This value identifies whether the LU is dependent or independent."

::= { appcLluOperEntry 2 }

appcLluOperLocalAddress OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (1))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The local address for this LU is a byte with a value ranging from 0 to 254. For dependent LUs, this value ranges from 1 to 254; for independent LUs this value is always 0."

::= { appcLluOperEntry 3 }

appcLluOperSessLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum number of sessions supported by this LU."

::= { appcLluOperEntry 4 }

appcLluOperBindRspMayQ OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates whether or not the local LU, as the sender of a BIND request, allows a partner LU to delay sending the BIND response if the partner LU cannot process the BIND request immediately."

::= { appcLluOperEntry 5 }

appcLluOperCompression OBJECT-TYPE

```

SYNTAX INTEGER {
    prohibited(1),
    required(2),
    negotiable(3)
}

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether compression is supported. The local LU uses this value for negotiation during session activation (SNA BIND).

- prohibited - specifies that no compression is to be used.
- required - specifies that compression is required.
- negotiable - specifies that the usage of compression is to be negotiated between the LUs. The level of compression is also negotiated."

```
 ::= { appcLluOperEntry 6 }
```

appcLluOperInBoundCompLevel OBJECT-TYPE

```

SYNTAX INTEGER {
    none(1),
    rle(2),
    lz9(3),
    lz10(4),
    lz12(5)
}

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the maximum level of compression supported for inbound data. The local LU uses this value in conjunction with appcLluOperCompression for negotiation during session activation (SNA BIND).

- none - specifies that no compression is to be used.
- rle - specifies run-length encoding compression in which a 1 or 2 byte sequence substitution is used for each repeated run of the same character.
- lz9 - specifies Lempel-Ziv-like compression in which 9 bit codes are used to substitute repeated substrings in the data stream. These codes are indices that refer to entries in a common dictionary generated adaptively at both sender and receiver as the data flows and compression occurs. The larger of number bits used for the code, the

more storage space is required for the dictionary,
but the larger the compression ratio.

- lz10 - specifies a 10 bit code Lempel-Ziv-like compression.
- lz12 - specifies a 12 bit code Lempel-Ziv-like compression."

::= { appcLluOperEntry 7 }

appcLluOperOutBoundCompLevel OBJECT-TYPE

SYNTAX INTEGER {
 none(1),
 rle(2),
 lz9(3),
 lz10(4),
 lz12(5)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the maximum level of compression supported for outbound data. The local LU uses this value in conjunction with appcLluAdminCompression for negotiation during session activation (SNA BIND).

- none - specifies that no compression is to be used.
- rle - specifies run-length encoding compression in which a 1 or 2 byte sequence substitution is used for each repeated run of the same character.
- lz9 - specifies Lempel-Ziv-like compression in which 9 bit codes are used to substitute repeated substrings in the data stream. These codes are indices that refer to entries in a common dictionary generated adaptively at both sender and receiver as the data flows and compression occurs. The larger of number bits used for the code, the more storage space is required for the dictionary, but the larger the compression ratio.
- lz10 - specifies a 10 bit code Lempel-Ziv-like compression.
- lz12 - specifies a 12 bit code Lempel-Ziv-like compression."

::= { appcLluOperEntry 8 }

appcLluOperCompRleBeforeLZ OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether run-length encoding is to be applied to the data before applying Lempel-Ziv-like compression. The local LU uses this value for negotiation during session activation (SNA BIND). This parameter is only supported if LZ compression is used."

```
::= { appcLluOperEntry 9 }
```

appcLluOperAlias OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A local alias for the local LU. If not known or not applicable, this object contains a zero-length string."

```
::= { appcLluOperEntry 10 }
```

appcLluOperActiveSessions OBJECT-TYPE

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the total number of active APPC sessions for this LU."

```
::= { appcLluOperEntry 11 }
```

```
-- *****
-- APPC LU Pair Admin Table
-- Objects in this table contain default or expected configuration
-- values for 6.2 LU pairs. An LU pair consists of a local LU and
-- a partner LU, which may or may not be local.
-- *****
```

appcLuPairAdminTable OBJECT-TYPE

SYNTAX SEQUENCE OF AppcLuPairAdminEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"APPC Partner LU administrative Table"

```
::= { appcLu 3 }
```

appcLuPairAdminEntry OBJECT-TYPE

SYNTAX AppcLuPairAdminEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entry of APPC Partner LU Information Table.
It is indexed by the local and partner LU Names."

INDEX { appcLuPairAdminLocLuName,
 appcLuPairAdminParLuName }

::= { appcLuPairAdminTable 1 }

AppcLuPairAdminEntry ::= SEQUENCE {

appcLuPairAdminLocLuName	DisplayString,
appcLuPairAdminParLuName	DisplayString,
appcLuPairAdminParLuAlias	DisplayString,
appcLuPairAdminSessLimit	Integer32,
appcLuPairAdminSessSec	INTEGER,
appcLuPairAdminSecAccept	INTEGER,
appcLuPairAdminLinkObjId	InstancePointer,
appcLuPairAdminParaSessSup	INTEGER

}

appcLuPairAdminLocLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The SNA name of the local LU to which this partner LU definition applies. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

The reserved value '*ALL' indicates that the partner LU definition applies to all local LUs, and not just to a single local LU."

::= { appcLuPairAdminEntry 1 }

appcLuPairAdminParLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The SNA name of the partner LU.
This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present."

```
::= { appcLuPairAdminEntry 2 }
```

```
appcLuPairAdminParLuAlias OBJECT-TYPE
```

```
SYNTAX DisplayString (SIZE (0..8))
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"A local alias for the partner LU. If not known or
not applicable, this object contains a zero-length
string."
```

```
::= { appcLuPairAdminEntry 3 }
```

```
appcLuPairAdminSessLimit OBJECT-TYPE
```

```
SYNTAX Integer32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The maximum number of sessions supported by this partner LU."
```

```
::= { appcLuPairAdminEntry 4 }
```

```
appcLuPairAdminSessSec OBJECT-TYPE
```

```
SYNTAX INTEGER {
    required(1),
    accepted(2),
    notAllowed(3)
}
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Specifies the type of session-level security information that
a local LU will accept on BIND requests it receives from the
partner LU."
```

- required - Specifies that the BIND request must carry session level verification information that will be verified upon receipt.
- accepted - Specifies that the BIND request may carry session level verification information that will be verified upon receipt.
- notAllowed - Specifies that the BIND request must not carry session level verification information."

```
::= { appcLuPairAdminEntry 5 }
```

```
appcLuPairAdminSecAccept OBJECT-TYPE
```

```
SYNTAX INTEGER {
```

```

        none(1),
        conversation(2),
        alreadyVerified(3),
        persistentVerification(4),
        aVandpV(5)
    }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Specifies support for different levels of access security
    information in ATTACH requests received from this partner LU.

```

Possible values are:

- none - No access security information will be accepted on allocation requests (ATTACH) from this LU.
- conversation - Allocation requests will not be accepted that include already verified or persistent verification indicators. Accept conversation-level access security information, which must include both a user Id and password, and may also include a profile.
- alreadyVerified - Allocation requests will be accepted that include already verified indicators. Persistent verification indicators will not be accepted.
- persistentVerification - Allocation requests will be accepted that include persistent verification indicators. Already verified indicators will not be accepted.
- aVandpV - Allocation requests will be accepted that include already verified or persistent verification indicators."

```
 ::= { appcLuPairAdminEntry 6 }
```

```

appcLuPairAdminLinkObjId OBJECT-TYPE
    SYNTAX InstancePointer
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

```

"Specifies the link associated with this partner LU. This value points to the row in the table containing information on

the link instance. (e.g., the `sdlcLSAdminTable` of the SNA DLC MIB module). This object may be NULL if the link is not specified or if a link is not applicable (as for APPN-level nodes)."

```
::= { appcLuPairAdminEntry 7 }
```

```
appcLuPairAdminParaSessSup OBJECT-TYPE
```

```
SYNTAX INTEGER {
    no(1),
    yes(2)
}
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Defined Parallel Sessions Supported.
```

Indicates whether or not multiple sessions between the partner LU and its associated local LU are permitted. Parallel session support also indicates that Change Number of Sessions (CNOS) will be used to negotiate session limits between the LUs."

```
::= { appcLuPairAdminEntry 8 }
```

```
-- *****
-- APPC LU Pair Oper Table
-- Objects in this table contain current operational values, such
-- as state values or negotiated parameters, for 6.2 LU pairs.
-- *****
```

```
appcLuPairOperTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF AppcLuPairOperEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Table of active partner/local LU pairs. Two entries are
present in the table when both LUs in a pair are local."
```

```
::= { appcLu 4 }
```

```
appcLuPairOperEntry OBJECT-TYPE
```

```
SYNTAX AppcLuPairOperEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Entry representing one partner/local LU pair."
```

```
INDEX { appcLuPairOperLocLuName,
```

```

        appcLuPairOperParLuName  }

 ::= { appcLuPairOperTable 1 }

AppcLuPairOperEntry ::= SEQUENCE {
    appcLuPairOperLocLuName      DisplayString,
    appcLuPairOperParLuName      DisplayString,
    appcLuPairOperParLuAlias     DisplayString,
    appcLuPairOperSessLimit      Integer32,
    appcLuPairOperSessSec        INTEGER,
    appcLuPairOperSecAccept      INTEGER,
    appcLuPairOperLinkObjId      InstancePointer,
    appcLuPairOperParaSessSup     INTEGER,
    appcLuPairOperParaSessSupLS  INTEGER,
    appcLuPairOperState          INTEGER
}

appcLuPairOperLocLuName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..17))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The SNA name of the local LU.  This field is from 1 to 17
        characters in length, including a period (.) which separates
        the NetId from the NAU name if the NetId is present.

        If this object has the same value as appcLluOperName,
        then the two entries being indexed apply to the same
        resource (specifically, to the same local LU)."
```

```

 ::= { appcLuPairOperEntry 1 }

appcLuPairOperParLuName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..17))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The SNA name of the partner LU.
        This field is from 1 to 17 characters in
        length, including a period (.) which separates the
        NetId from the NAU name if the NetId is present."
```

```

 ::= { appcLuPairOperEntry 2 }

appcLuPairOperParLuAlias OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..8))
    MAX-ACCESS read-only
    STATUS current

```


DESCRIPTION

"A local alias for the partner LU. If not known or not applicable, this object contains a zero-length string."

::= { appcLuPairOperEntry 3 }

appcLuPairOperSessLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum number of sessions supported by this partner LU."

::= { appcLuPairOperEntry 4 }

appcLuPairOperSessSec OBJECT-TYPE

SYNTAX INTEGER {
 required(1),
 accepted(2),
 notAllowed(3)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the type of security information that a local LU will accept on BIND requests it receives from the partner LU."

- required - Specifies that the BIND request must carry session level verification information that will be verified upon receipt.
- accepted - Specifies that the BIND request may carry session level verification information that will be verified upon receipt.
- notAllowed - Specifies that the BIND request must not carry session level verification information."

::= { appcLuPairOperEntry 5 }

appcLuPairOperSecAccept OBJECT-TYPE

SYNTAX INTEGER {
 none(1),
 conversation(2),
 alreadyVerified(3),
 persistentVerification(4),
 aVandpV(5)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies support for different levels of security acceptance information in ATTACH requests received from this partner LU.

Possible values are:

- none - No access security information will be accepted on allocation requests (ATTACH) from this LU.
- conversation - Allocation requests will not be accepted that include already verified or persistent verification indicators. Accept conversation-level access security information, which must include both a user Id and password, and may also include a profile.
- alreadyVerified - Allocation requests will be accepted that include already verified indicators. Persistent verification indicators will not be accepted.
- persistentVerification - Allocation requests will be accepted that include persistent verification indicators. Already verified indicators will not be accepted.
- aVandpV - Allocation requests will be accepted that include already verified or persistent verification indicators."

::= { appcLuPairOperEntry 6 }

appcLuPairOperLinkObjId OBJECT-TYPE

SYNTAX InstancePointer

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the link associated with this partner LU. This value points to the row in the table containing information on the link instance. (e.g., the sdlcLSAdminTable of the SNA DLC MIB module). This object may be NULL if the link is not specified or if a link is not applicable (as for APPN-level nodes)."

::= { appcLuPairOperEntry 7 }

appcLuPairOperParaSessSup OBJECT-TYPE

```
SYNTAX INTEGER {
    no(1),
    yes(2)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Active Parallel Sessions Supported.

Indicates whether or not multiple session between the partner LU and its associated local LU are permitted. Parallel session support also indicates that Change Number of Sessions (CNOS) will be used to negotiate session limits between the LUs."

```
::= { appcLuPairOperEntry 8 }
```

appcLuPairOperParaSessSupLS OBJECT-TYPE

```
SYNTAX INTEGER {
    no(1),
    yes(2)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Active Parallel Sessions Supported - last starting value.

This object represents the initial value proposed by the local LU the last time this capability was negotiated, i.e., when the first session was bound between the local LU and its partner."

```
::= { appcLuPairOperEntry 9 }
```

appcLuPairOperState OBJECT-TYPE

```
SYNTAX INTEGER {
    inactive (1),
    active (2)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value identifies the current operational state of this LU pair:

inactive (1) - no active or pending session exists
between the LUs.

active (2) - an active or pending session exists

between the LUs."

```
::= { appcLuPairOperEntry 10 }
```

```
-- *****
-- APPC Mode Admin Table
-- Objects in this table contain default or expected configuration
-- values for session modes.
-- Modes that have active sessions appear in the appcModeOperTable.
-- *****
```

```
appcModeAdminTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AppcModeAdminEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "APPC Mode Table"
```

```
::= { appcLu 5 }
```

```
appcModeAdminEntry OBJECT-TYPE
    SYNTAX AppcModeAdminEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Entry of APPC Mode Information Table."
    INDEX { appcModeAdminLocLuName,
            appcModeAdminParLuName,
            appcModeAdminModeName }
```

```
::= { appcModeAdminTable 1 }
```

```
AppcModeAdminEntry ::= SEQUENCE {
    appcModeAdminLocLuName      DisplayString,
    appcModeAdminParLuName      DisplayString,
    appcModeAdminModeName       DisplayString,
    appcModeAdminCosName        DisplayString,
    appcModeAdminSessEndTpName   DisplayString,
    appcModeAdminMaxSessLimit    Integer32,
    appcModeAdminMinCwinLimit    Integer32,
    appcModeAdminMinClosLimit    Integer32,
    appcModeAdminConWinAutoActLmt Integer32,
    appcModeAdminRecvPacWinSz    Integer32,
    appcModeAdminSendPacWinSz    Integer32,
    appcModeAdminPrefRecvRuSz    Integer32,
    appcModeAdminPrefSendRuSz    Integer32,
    appcModeAdminRecvRuSzUpBnd   Integer32,
    appcModeAdminSendRuSzUpBnd   Integer32,
```

```

appcModeAdminRecvRuSzLoBnd      Integer32,
appcModeAdminSendRuSzLoBnd      Integer32,
appcModeAdminSingSessReinit      INTEGER,
appcModeAdminCompression        INTEGER,
appcModeAdminInBoundCompLevel    INTEGER,
appcModeAdminOutBoundCompLevel   INTEGER,
appcModeAdminCompRleBeforeLZ     INTEGER,
appcModeAdminSyncLvl             INTEGER,
appcModeAdminCrypto              INTEGER
    }

```

appcModeAdminLocLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The SNA name of the local LU to which this mode definition applies. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

The reserved value '*ALL' indicates that the mode definition applies to all local LUs for the SNA node identified by appcLocalCpName, and not just to a single local LU."

::= { appcModeAdminEntry 1 }

appcModeAdminParLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The SNA name of the partner LU to which this mode definition applies. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

The reserved value '*ALL' indicates that the mode definition applies to all partner LUs for the SNA node identified by appcModeAdminLocLuName, and not just to a single partner LU."

::= { appcModeAdminEntry 2 }

appcModeAdminModeName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..8))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Specifies the mode name. A mode defines the characteristics for a group of sessions. The mode name can be blank (8 space characters). "

::= { appcModeAdminEntry 3 }

appcModeAdminCosName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the class of service (COS) name associated with this mode. If the implementation does not support COS names, a null string is returned."

::= { appcModeAdminEntry 4 }

appcModeAdminSessEndTpName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..64))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the name of the transaction program (TP) to be invoked when a session using this mode is deactivated or ended. If no such TP is defined, this object is a null string. When the TP name consists entirely of displayable EBCDIC code points, it is mapped directly to the equivalent ASCII display string. However, registered TP names always have a non-displayable EBCDIC code point (value less than or equal to x'3F') as the first character, so they cannot be directly mapped to an ASCII display string. These TP names are converted to a display string that is equivalent to a hexadecimal display of the EBCDIC code points. For example, the 2-byte TP name x'06F1' (CNOS) is converted to the 6-byte ASCII display string '06F1' (including the two single quotes)."

::= { appcModeAdminEntry 5 }

appcModeAdminMaxSessLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the maximum value that the local LU is to use, during CNOS processing, for the session limit. The local LU, as a target LU, will negotiate a higher session limit if

receives in the CNOS request down to this maximum value. The local LU, as a source LU, will restrict the session limit it sends in a CNOS request to a value less than or equal to this maximum value."

::= { appcModeAdminEntry 6 }

appcModeAdminMinCwinLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the default minimum contention winner sessions limit. Some implementations internally issue a INITIALIZE_SESSION_LIMIT verb when a Mode is created. This value is the parameter used for the CNOS processing of that verb. This parameter is not used when issuing an explicit INITIALIZE_SESSION_LIMIT verb. The equivalent object in appcCnosCommandTable is used."

::= { appcModeAdminEntry 7 }

appcModeAdminMinClosLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the default minimum contention loser sessions limit. Some implementations internally issue a INITIALIZE_SESSION_LIMIT verb when a Mode is created. This value is the parameter used for the CNOS processing of that verb. This is the same as target minimum contention winner sessions. This parameter is not used when issuing an explicit INITIALIZE_SESSION_LIMIT verb. The equivalent object in appcCnosCommandTable is used."

::= { appcModeAdminEntry 8 }

appcModeAdminConWinAutoActLmt OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the limit on the number of contention winner sessions to be automatically activated when the minimum number of contention winner sessions increases (as a result of CNOS processing). The actual number of sessions activated is the lesser of this value and the new minimum number of contention winner sessions. "

::= { appcModeAdminEntry 9 }

appcModeAdminRecvPacWinSz OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the size of the receive pacing window. This value is used for negotiation during session activations (SNA BIND).

The meaning of this value when set to 0 depends on whether adaptive pacing is supported:

adaptive pacing	No limit on window size
fixed pacing	No pacing is supported"

::= { appcModeAdminEntry 10 }

appcModeAdminSendPacWinSz OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the size of the send pacing window. This value is used for negotiation during session activations (SNA BIND).

The meaning of this value when set to 0 depends on whether adaptive pacing is supported:

adaptive pacing	No limit on window size
fixed pacing	No pacing is supported"

::= { appcModeAdminEntry 11 }

appcModeAdminPrefRecvRuSz OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the preferred receive RU (Request Unit) size of normal-flow requests on the sessions. This value must be less than or equal to the value specified in appcModeAdminRecvRuSzUpBnd and greater than or equal to the value specified in appcModeAdminRecvRuSzLoBnd.

The local LU specifies this value for the receive maximum RU size in session activation (SNA BIND) requests and responses. It will allow negotiation up to the appcModeAdminRecvRuSzUpBnd value or down to the appcModeAdminRecvRuSzLoBnd value."


```
::= { appcModeAdminEntry 12 }
```

appcModeAdminPrefSendRuSz OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the preferred send RU (Request Unit) size of normal-flow requests on the sessions. This value must be less than or equal to the value specified in appcModeAdminSendRuSzUpBnd and greater than or equal to the value specified in appcModeAdminSendRuSzLoBnd.

The local LU specifies this value for the send maximum RU size in session activation (SNA BIND) requests and responses. It will allow negotiation up to the appcModeAdminSendRuSzUpBnd value or down to the appcModeAdminSendRuSzLoBnd value."

```
::= { appcModeAdminEntry 13 }
```

appcModeAdminRecvRuSzUpBnd OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the upper bound for the maximum receive RU (Request Unit) size of normal-flow requests. This is used for negotiation during session activations (SNA BIND). "

```
::= { appcModeAdminEntry 14 }
```

appcModeAdminSendRuSzUpBnd OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the upper bound for the maximum send RU (Request Unit) size of normal-flow requests. This is used for negotiation during session activations (SNA BIND). "

```
::= { appcModeAdminEntry 15 }
```

appcModeAdminRecvRuSzLoBnd OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the lower bound for the maximum receive RU (Request

Unit) size of normal-flow requests. This is used for negotiation during session activations (SNA BIND). "

::= { appcModeAdminEntry 16 }

appcModeAdminSendRuSzLoBnd OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the lower bound for the maximum send RU (Request Unit) size of normal-flow requests. This is used for negotiation during session activations (SNA BIND). "

::= { appcModeAdminEntry 17 }

appcModeAdminSingSessReinit OBJECT-TYPE

SYNTAX INTEGER {
 notApplicable(1),
 operatorControlled(2),
 primaryOnly(3),
 secondaryOnly(4),
 primaryOrSecondary(5)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the responsibility for session reinitiation of a single session with the partner LU (when the session goes down). The local LU uses this parameter to specify the session reinitiation responsibility in session activation (SNA BIND) requests and responses.

- | | |
|--------------------|--|
| notApplicable | - specifies that this parameter has no meaning since the value of appcLuPairAdminParaSessSup is yes. The field in the SNA BIND is reserved (set to zero). |
| operatorControlled | - specifies that neither LU will automatically attempt to reinitiate the session. The operator on either side will manually reactivate the session. A contention race (both side reinitiating at the same time) is won by the LU with the lexicographically greater fully-qualified LU name. |
| primaryOnly | - specifies that the primary LU will |

```

                                automatically attempt to reinitiate
                                the session.
        secondaryOnly          - specifies that the secondary LU will
                                automatically attempt to reinitiate
                                the session.
        primaryOrSecondary    - specifies that either the primary or
                                the secondary may automatically
                                attempt to reinitiate the session.
                                A contention race is handled the
                                same way as with operatorControlled.
    "
 ::= { appcModeAdminEntry 18 }

appcModeAdminCompression OBJECT-TYPE
    SYNTAX INTEGER {
        prohibited(1),
        required(2),
        negotiable(3)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Specifies whether compression is supported.  The local LU uses
        this value for negotiation during session activation (SNA
        BIND).

        prohibited - specifies that no compression is to be used.
        required   - specifies that compression is required.
        negotiable  - specifies that the usage of compression
                     is to be negotiated between the LUs. The
                     level of compression is also negotiated."

 ::= { appcModeAdminEntry 19 }

appcModeAdminInBoundCompLevel OBJECT-TYPE
    SYNTAX INTEGER {
        none(1),
        rle(2),
        lz9(3),
        lz10(4),
        lz12(5)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Specifies the maximum level of compression supported for
        inbound data.  The local LU uses this value in conjunction with
        appcModeAdminCompression for negotiation during session

```

activation (SNA BIND).

- none - specifies that no compression is to be used.
- rle - specifies run-length encoding compression in which a 1 or 2 byte sequence substitution is used for each repeated run of the same character.
- lz9 - specifies Lempel-Ziv-like compression in which 9 bit codes are used to substitute repeated substrings in the data stream. These codes are indices that refer to entries in a common dictionary generated adaptively at both sender and receiver as the data flows and compression occurs. The larger of number bits used for the code, the more storage space is required for the dictionary, but the larger the compression ratio.
- lz10 - specifies a 10 bit code Lempel-Ziv-like compression.
- lz12 - specifies a 12 bit code Lempel-Ziv-like compression."

::= { appcModeAdminEntry 20 }

appcModeAdminOutBoundCompLevel OBJECT-TYPE

```
SYNTAX INTEGER {
    none(1),
    rle(2),
    lz9(3),
    lz10(4),
    lz12(5)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the maximum level of compression supported for outbound data. The local LU uses this value in conjunction with appcModeAdminCompression for negotiation during session activation (SNA BIND).

- none - specifies that no compression is to be used.
- rle - specifies run-length encoding compression in which a 1 or 2 byte sequence substitution is used for each repeated run of the same character.
- lz9 - specifies Lempel-Ziv-like compression in which 9 bit codes are used to substitute repeated substrings in the data stream. These codes are indices that refer to entries in a common dictionary generated adaptively at both sender and receiver as the data flows and compression occurs. The larger of number bits used for the code, the more storage space is required for the dictionary,

but the larger the compression ratio.
 lz10 - specifies a 10 bit code Lempel-Ziv-like compression.
 lz12 - specifies a 12 bit code Lempel-Ziv-like compression."

::= { appcModeAdminEntry 21 }

appcModeAdminCompRleBeforeLZ OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether run-length encoding is to be applied to the data before applying Lempel-Ziv-like compression. The local LU uses this value for negotiation during session activation (SNA BIND). This parameter is only supported if LZ compression is used."

::= { appcModeAdminEntry 22 }

appcModeAdminSyncLvl OBJECT-TYPE

SYNTAX INTEGER {
 none(1),
 noneConfirm(2),
 noneConfirmSyncPoint(3)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the sync level support. This value is used for negotiation during session activations (SNA BIND).

none	- No sync level is supported.
noneConfirm	- None and Confirm levels supported.
noneConfirmSyncPoint	- None, Confirm, and Sync Point is supported.

"

::= { appcModeAdminEntry 23 }

appcModeAdminCrypto OBJECT-TYPE

SYNTAX INTEGER {
 notSupported(1),
 mandatory(2),
 selective(3)
 }

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"Specifies whether session-level cryptography is supported.
 This value is used for negotiation during session activations
 (SNA BIND).

notSupported	-	Specifies session-level cryptography is not to be used.
mandatory	-	Specifies session-level cryptography must be used.
selective	-	Specifies session-level cryptography is required just on selected requests flowing on the sessions."

::= { appcModeAdminEntry 24 }

```
-- *****
-- APPC Mode Oper Table
-- Objects in this table contain current operational values, such
-- as state values or negotiated parameters, for session modes.
--
```

```
-- *****
```

appcModeOperTable OBJECT-TYPE

SYNTAX SEQUENCE OF AppcModeOperEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Operational APPC Mode Information. Two entries are present in the table when both LUs in a pair are local."

::= { appcLu 6 }

appcModeOperEntry OBJECT-TYPE

SYNTAX AppcModeOperEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entry of APPC mode operational information table. This entry does not augment the appcModeAdminEntry, but reflects an actual operational mode for a given local LU - partner LU pair."

INDEX { appcModeOperLocLuName,
 appcModeOperParLuName,
 appcModeOperModeName }

::= { appcModeOperTable 1 }

```

AppcModeOperEntry      ::= SEQUENCE {

    appcModeOperLocLuName      DisplayString,
    appcModeOperParLuName      DisplayString,
    appcModeOperModeName       DisplayString,
    appcModeOperCosName        DisplayString,
    appcModeOperSessEndTpName   DisplayString,
    appcModeOperSessLimit      Integer32,
    appcModeOperMaxSessLimit    Integer32,
    appcModeOperMinCwinLimit    Integer32,
    appcModeOperMinClosLimit    Integer32,
    appcModeOperConWinAutoActLmt Integer32,
    appcModeOperRecvPacWinSz    Integer32,
    appcModeOperSendPacWinSz    Integer32,
    appcModeOperPrefRecvRuSz    Integer32,
    appcModeOperPrefSendRuSz    Integer32,
    appcModeOperRecvRuSzUpBnd   Integer32,
    appcModeOperSendRuSzUpBnd   Integer32,
    appcModeOperRecvRuSzLoBnd   Integer32,
    appcModeOperSendRuSzLoBnd   Integer32,
    appcModeOperSingSessReinit  INTEGER,
    appcModeOperCompression     INTEGER,
    appcModeOperInBoundCompLevel INTEGER,
    appcModeOperOutBoundCompLevel INTEGER,
    appcModeOperCompRleBeforeLZ  INTEGER,
    appcModeOperSyncLvl         INTEGER,
    appcModeOperCrypto          INTEGER,
    appcModeOperSyncLvlLastStart INTEGER,
    appcModeOperCryptoLastStart  INTEGER,
    appcModeOperCNOSNeg         INTEGER,
    appcModeOperActCwin         Gauge32,
    appcModeOperActClos         Gauge32,
    appcModeOperPndCwin         Gauge32,
    appcModeOperPndClos         Gauge32,
    appcModeOperPtmCwin         Gauge32,
    appcModeOperPtmClos         Gauge32,
    appcModeOperDrainSelf        INTEGER,
    appcModeOperDrainPart        INTEGER
}

appcModeOperLocLuName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..17))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The SNA name of the local LU.  This field is from 1 to 17
        characters in length, including a period (.) which separates
        the NetId from the NAU name if the NetId is present."

```

If this object has the same value as appcLluOperName, then the two entries being indexed apply to the same resource (specifically, to the same local LU)."

::= { appcModeOperEntry 1 }

appcModeOperParLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The SNA name of the partner LU. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

If this object has the same value as appcLuPairOperParLuName, then the two entries being indexed apply to the same resource (specifically, to the same partner LU)."

::= { appcModeOperEntry 2 }

appcModeOperModeName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..8))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Specifies the mode name. A mode defines the characteristics for a group of sessions. The mode name can be blank (8 space characters). "

::= { appcModeOperEntry 3 }

appcModeOperCosName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the class of service (COS) name associated with this mode. If the implementation does not support COS names, a zero-length string is returned."

::= { appcModeOperEntry 4 }

appcModeOperSessEndTpName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..64))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the name of the transaction program (TP) to be invoked when a session using this mode is deactivated or ended. If the name is NULL, no transaction program is invoked. When the TP name consists entirely of displayable EBCDIC code points, it is mapped directly to the equivalent ASCII display string. However, registered TP names always have a non-displayable EBCDIC code point (value less than or equal to x'3F') as the first character, so they cannot be directly mapped to an ASCII display string. These TP names are converted to a display string that is equivalent to a hexadecimal display of the EBCDIC code points. For example, the 2-byte TP name x'06F1' (CNOS) is converted to the 6-byte ASCII display string '06F1' (including the two single quotes)."

::= { appcModeOperEntry 5 }

appcModeOperSessLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the current session limit of this mode as negotiated through APPC CNOS (Change Number of Sessions) processing. Identifies the total number of sessions that can be established between the local and partner LU using this mode."

::= { appcModeOperEntry 6 }

appcModeOperMaxSessLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the maximum value that the local LU is to use, during CNOS processing, for the session limit. The local LU, as a target LU, will negotiate a higher session limit it receives in the CNOS request down to this maximum value. The local LU, as a source LU, will restrict the session limit it sends in a CNOS request to a value less than or equal to this maximum value."

::= { appcModeOperEntry 7 }

appcModeOperMinCwinLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the minimum contention winner sessions limit that was negotiated via CNOS processing."

::= { appcModeOperEntry 8 }

appcModeOperMinClosLimit OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the minimum contention loser sessions limit that was negotiated via CNOS processing. This is the same as target minimum contention winner sessions."

::= { appcModeOperEntry 9 }

appcModeOperConWinAutoActLmt OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the limit on the number of contention winner sessions to be automatically activated when the minimum number of contention winner sessions increases (as a result of CNOS processing). The actual number of sessions activated is the lesser of this value and the new minimum number of contention winner sessions. "

::= { appcModeOperEntry 10 }

appcModeOperRecvPacWinSz OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the size of the receive pacing window. This value is used for negotiation during session activations (SNA BIND).

The meaning of this value when set to 0 depends on whether adaptive pacing is supported:

adaptive pacing	-	No limit on window size
fixed pacing	-	No pacing is supported"

::= { appcModeOperEntry 11 }

appcModeOperSendPacWinSz OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the size of the send pacing window. This value is used for negotiation during session activations (SNA BIND).

The meaning of this value when set to 0 depends on whether adaptive pacing is supported:

adaptive pacing	No limit on window size
fixed pacing	No pacing is supported"

::= { appcModeOperEntry 12 }

appcModeOperPrefRecvRuSz OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the preferred receive RU (Request Unit) size of normal-flow requests on the sessions. This value must be less than or equal to the value specified in appcModeOperRecvRuSzUpBnd and greater than or equal to the value specified in appcModeOperRecvRuSzLoBnd.

The local LU specifies this value for the receive maximum RU size in session activation (SNA BIND) requests and responses. It will allow negotiation up to the appcModeOperRecvRuSzUpBnd value or down to the appcModeOperRecvRuSzLoBnd value."

::= { appcModeOperEntry 13 }

appcModeOperPrefSendRuSz OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the preferred send RU (Request Unit) size of normal-flow requests on the sessions. This value must be less than or equal to the value specified in appcModeOperSendRuSzUpBnd and greater than or equal to the value specified in appcModeOperSendRuSzLoBnd.

The local LU specifies this value for the send maximum RU size in session activation (SNA BIND) requests and responses. It will allow negotiation up to the appcModeOperSendRuSzUpBnd value or down to the appcModeOperSendRuSzLoBnd value."

::= { appcModeOperEntry 14 }

appcModeOperRecvRuSzUpBnd OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the upper bound for the maximum receive RU (Request Unit) size of normal-flow requests. This is used for negotiation during session activations (SNA BIND). "

::= { appcModeOperEntry 15 }

appcModeOperSendRuSzUpBnd OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the upper bound for the maximum send RU (Request Unit) size of normal-flow requests. This is used for negotiation during session activations (SNA BIND). "

::= { appcModeOperEntry 16 }

appcModeOperRecvRuSzLoBnd OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the lower bound for the maximum receive RU (Request Unit) size of normal-flow requests. This is used for negotiation during session activations (SNA BIND). "

::= { appcModeOperEntry 17 }

appcModeOperSendRuSzLoBnd OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the lower bound for the maximum send RU (Request Unit) size of normal-flow requests. This is used for negotiation during session activations (SNA BIND). "

::= { appcModeOperEntry 18 }

appcModeOperSingSessReinit OBJECT-TYPE

SYNTAX INTEGER {
 notApplicable(1),
 operatorControlled(2),

```

        primaryOnly(3),
        secondaryOnly(4),
        primaryOrSecondary(5)
    }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Specifies the responsibility for session reinitiation of a
    single session with the partner LU (when the session goes
    down). The local LU uses this parameter to specify the session
    reinitiation responsibility in session activation (SNA BIND)
    requests and responses.

        notApplicable      - specifies that this parameter has
                             no meaning since the value of
                             appcLuPairOperParaSessSup is yes.
                             The field in the SNA BIND is
                             reserved (set to zero).

        operatorControlled - specifies that neither LU will
                             automatically attempt to reinitiate
                             the session. The operator on either
                             side will manually reactivate the
                             session. A contention race (both
                             side reinitiating at the same time)
                             is won by the LU with the
                             lexicographically greater fully-
                             qualified LU name.

        primaryOnly        - specifies that the primary LU will
                             automatically attempt to reinitiate
                             the session.

        secondaryOnly      - specifies that the secondary LU will
                             automatically attempt to reinitiate
                             the session.

        primaryOrSecondary - specifies that either the primary or
                             the secondary may automatically
                             attempt to reinitiate the session.
                             A contention race is handled the
                             same way as with operatorControlled.

    "
 ::= { appcModeOperEntry 19 }

appcModeOperCompression OBJECT-TYPE
    SYNTAX INTEGER {
        prohibited(1),
        required(2),
        negotiable(3)
    }
MAX-ACCESS read-only

```

STATUS current

DESCRIPTION

"Specifies whether compression is supported. The local LU uses this value for negotiation during session activation (SNA BIND).

prohibited - specifies that no compression is to be used.
 required - specifies that compression is required.
 negotiable - specifies that the usage of compression is to be negotiated between the LUs. The level of compression is also negotiated."

::= { appcModeOperEntry 20 }

appcModeOperInBoundCompLevel OBJECT-TYPE

SYNTAX INTEGER {
 none(1),
 rle(2),
 lz9(3),
 lz10(4),
 lz12(5)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the maximum level of compression supported for inbound data. The local LU uses this value in conjunction with appcModeOperCompression for negotiation during session activation (SNA BIND).

none - specifies that no compression is to be used.
 rle - specifies run-length encoding compression in which a 1 or 2 byte sequence substitution is used for each repeated run of the same character.
 lz9 - specifies Lempel-Ziv-like compression in which 9 bit codes are used to substitute repeated substrings in the data stream. These codes are indices that refer to entries in a common dictionary generated adaptively at both sender and receiver as the data flows and compression occurs. The larger of number bits used for the code, the more storage space is required for the dictionary, but the larger the compression ratio.
 lz10 - specifies a 10 bit code Lempel-Ziv-like compression.
 lz12 - specifies a 12 bit code Lempel-Ziv-like compression."

::= { appcModeOperEntry 21 }

appcModeOperOutBoundCompLevel OBJECT-TYPE

```

SYNTAX INTEGER {
    none(1),
    rle(2),
    lz9(3),
    lz10(4),
    lz12(5)
}

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the maximum level of compression supported for outbound data. The local LU uses this value in conjunction with appcModeOperCompression for negotiation during session activation (SNA BIND).

- none - specifies that no compression is to be used.
- rle - specifies run-length encoding compression in which a 1 or 2 byte sequence substitution is used for each repeated run of the same character.
- lz9 - specifies Lempel-Ziv-like compression in which 9 bit codes are used to substitute repeated substrings in the data stream. These codes are indices that refer to entries in a common dictionary generated adaptively at both sender and receiver as the data flows and compression occurs. The larger of number bits used for the code, the more storage space is required for the dictionary, but the larger the compression ratio.
- lz10 - specifies a 10 bit code Lempel-Ziv-like compression.
- lz12 - specifies a 12 bit code Lempel-Ziv-like compression."

```
 ::= { appcModeOperEntry 22 }
```

appcModeOperCompRleBeforeLZ OBJECT-TYPE

```

SYNTAX INTEGER {
    no(1),
    yes(2)
}

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether run-length encoding is to be applied to the data before applying Lempel-Ziv-like compression. The local LU uses this value for negotiation during session activation (SNA BIND). This parameter is only supported if LZ compression is used."

```
::= { appcModeOperEntry 23 }
```

```
appcModeOperSyncLvl OBJECT-TYPE
```

```
SYNTAX INTEGER {
    none(1),
    noneConfirm(2),
    noneConfirmSyncPoint(3)
}
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Specifies the sync level support for sessions involving this
LU pair and mode name.
```

```

    none                -      No sync level is supported.
    noneConfirm          -      None and Confirm level supported.
    noneConfirmSyncPoint -      None, Confirm and Sync Point is
                                supported.
"
```

```
::= { appcModeOperEntry 24 }
```

```
appcModeOperCrypto OBJECT-TYPE
```

```
SYNTAX INTEGER {
    notSupported(1),
    mandatory(2),
    selective(3)
}
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Specifies whether session-level cryptography is supported for
sessions involving this LU pair and mode name.
```

```

    notSupported      -      Specifies session-level cryptography
                             is not being used.
    mandatory          -      Specifies session-level cryptography
                             in being used on all requests
                             flowing on the sessions.
    selective          -      Specifies session-level cryptography
                             is required just on selected
                             requests flowing on the sessions."
"
```

```
::= { appcModeOperEntry 25 }
```

```
appcModeOperSyncLvlLastStart OBJECT-TYPE
```

```
SYNTAX INTEGER {
    none(1),
```



```

        noneConfirm(2),
        noneConfirmSyncPoint(3)
    }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Specifies the sync level support.  This value represents the
    initial value proposed by the local LU the last time this
    capability was negotiated, i.e., when the first session was
    bound between the local LU and its partner.

        none           -      No sync level is supported.
        noneConfirm    -      None and Confirm level supported.
        noneConfirmSyncPoint - None, Confirm and Sync Point is
                                supported.

    "

 ::= { appcModeOperEntry 26 }

appcModeOperCryptoLastStart OBJECT-TYPE
    SYNTAX INTEGER {
        notSupported(1),
        mandatory(2),
        selective(3)
    }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Specifies whether session-level cryptography is supported.
    This value represents the initial value proposed by the local
    LU the last time this capability was negotiated, i.e., when
    the first session was bound between the local LU and its
    partner.

        notSupported    -      Specifies session-level cryptography
                                is not to be used.
        mandatory       -      Specifies session-level cryptography
                                must be used.
        selective       -      Specifies session-level cryptography
                                is required just on selected
                                requests flowing on the sessions."

 ::= { appcModeOperEntry 27 }

appcModeOperCNOSNeg OBJECT-TYPE
    SYNTAX INTEGER {
        no(1),
        yes(2)
    }

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Specifies whether CNOS negotiation is in process. CNOS negotiation is used to set or change the various session limits for a mode."

::= { appcModeOperEntry 28 }

appcModeOperActCwin OBJECT-TYPE
 SYNTAX Gauge32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Specifies the number of active contention winner sessions."

::= { appcModeOperEntry 29 }

appcModeOperActClos OBJECT-TYPE
 SYNTAX Gauge32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Specifies the number of active contention loser sessions."

::= { appcModeOperEntry 30 }

appcModeOperPndCwin OBJECT-TYPE
 SYNTAX Gauge32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Specifies the number of contention winner sessions that are pending activation."

::= { appcModeOperEntry 31 }

appcModeOperPndClos OBJECT-TYPE
 SYNTAX Gauge32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Specifies the number of contention loser sessions that are pending activation."

::= { appcModeOperEntry 32 }

appcModeOperPtmCwin OBJECT-TYPE

SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Specifies the number of contention winner sessions that are
 pending termination."

::= { appcModeOperEntry 33 }

appcModeOperPtmClos OBJECT-TYPE
 SYNTAX Gauge32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Specifies the number of contention loser sessions that are
 pending termination."

::= { appcModeOperEntry 34 }

appcModeOperDrainSelf OBJECT-TYPE
 SYNTAX INTEGER {
 no(1),
 yes(2)
 }
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Specifies whether the local LU is draining its conversations
 for this mode. When a mode session limit is reset (via a CNOS
 RESET_SESSION_LIMIT request), the local LU could be set to
 process all queued conversations before deactivating all of the
 sessions (using the SNA Bracket Initiation Stopped or BIS
 protocol). "

::= { appcModeOperEntry 35 }

appcModeOperDrainPart OBJECT-TYPE
 SYNTAX INTEGER {
 no(1),
 yes(2)
 }
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Specifies whether the partner LU is draining its conversations
 for this mode. When a mode session limit is reset (via a CNOS
 RESET_SESSION_LIMIT request), the partner LU could be set to
 process all queued conversations before deactivating all of the

sessions (using the SNA Bracket Initiation Stop or BIS protocol). "

::= { appcModeOperEntry 36 }

```
-- *****
-- APPC TP Admin Table
-- Objects in this table contain default or expected configuration
-- values for remotely attachable transaction programs.
-- *****
```

appcTpAdminTable OBJECT-TYPE
 SYNTAX SEQUENCE OF AppcTpAdminEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION
 "APPC Local TP Table"

::= { appcTp 1 }

appcTpAdminEntry OBJECT-TYPE
 SYNTAX AppcTpAdminEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION
 "Entry of APPC Local TP Information Table."

INDEX { appcTpAdminLocLuName,
 appcTpAdminTpName }

::= { appcTpAdminTable 1 }

AppcTpAdminEntry ::= SEQUENCE {
 appcTpAdminLocLuName DisplayString,
 appcTpAdminTpName DisplayString,
 appcTpAdminFileSpec DisplayString,
 appcTpAdminStartParm DisplayString,
 appcTpAdminTpOperation INTEGER,
 appcTpAdminInAttachTimeout Integer32,
 appcTpAdminRcvAllocTimeout Integer32,
 appcTpAdminSyncLvl INTEGER,
 appcTpAdminInstLmt Integer32,
 appcTpAdminStatus INTEGER,
 appcTpAdminLongRun INTEGER,
 appcTpAdminConvType INTEGER,
 appcTpAdminConvDuplex INTEGER,
 appcTpAdminConvSecReq INTEGER,
 appcTpAdminVerPip INTEGER,
 appcTpAdminPipSubNum Integer32

}

appcTpAdminLocLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The SNA name of the local LU to which this TP definition applies. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

The reserved value '*ALL' indicates that the TP definition applies to all local LUs, and not just to a single local LU."

::= { appcTpAdminEntry 1 }

appcTpAdminTpName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..64))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The local transaction program name. This name is sent on an ATTACH remote allocation request.

When the TP name consists entirely of displayable EBCDIC code points, it is mapped directly to the equivalent ASCII display string. However, registered TP names always have a non-displayable EBCDIC code point (value less than or equal to x'3F') as the first character, so they cannot be directly mapped to an ASCII display string. These TP names are converted to a display string that is equivalent to a hexadecimal display of the EBCDIC code points. For example, the 2-byte TP name x'06F1' (CNOS) is converted to the 6-byte ASCII display string '06F1' (including the two single quotes)."

::= { appcTpAdminEntry 2 }

appcTpAdminFileSpec OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..80))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The local file specification of the transaction program. May be a zero-length string if not applicable."

::= { appcTpAdminEntry 3 }

appcTpAdminStartParm OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A parameter string passed to the transaction program when it is started. May be a zero-length string if not supported. "

::= { appcTpAdminEntry 4 }

appcTpAdminTpOperation OBJECT-TYPE

SYNTAX INTEGER {

other(1),

queuedOperatorStarted(2),

queuedOperatorPreloaded(3),

queuedAmStarted(4),

nonqueuedAmStarted(5)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies how the program will be started.

other - Specifies that the program operation is none of the methods specified. It may be a product-specific method.

queuedOperatorStarted - Specifies that one version of the program will be run at a time. If an incoming attach arrives and the program has not been started yet, APPC will issue a message to the operator to start the specified program. Subsequent attaches that arrive while the program is active will be queued.

queuedOperatorPreloaded - Specifies that one version of the program will be run at a time. If an incoming attach arrives and the program has not been started yet, the Attach will be rejected. The APPC attach manager determines that a TP has started upon reception of an APPC RECEIVE_ALLOCATE verb, or a CPI-C Accept_Conversation (CMACCP) or Specify_Local_TP_Name (CMSLTP) call. No message is sent to the operator. Subsequent attaches that arrive while the program is active are queued.

queuedAmStarted - Specifies that one version of the program will be run at a time and will be started by the APPC attach manager. Subsequent attaches

that arrive while the program is active will be queued.

nonqueuedAmStarted - Specifies that multiple copies of the program will be run at a time and will be started by the APPC attach manager."

::= { appcTpAdminEntry 5 }

appcTpAdminInAttachTimeout OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the number of seconds incoming attaches will be queued waiting for an APPC program to issue a RECEIVE_ALLOCATE verb or for a CPI-C program to issue an Accept_Conversation (CMACCP) call. This parameter is meaningful only when appcTpAdminTpOperation has one of the following values:

 queuedOperatorStarted
 queuedOperatorPreloaded
 queuedAmStarted

A value of zero indicates that there is no timeout."

::= { appcTpAdminEntry 6 }

appcTpAdminRcvAllocTimeout OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the number of seconds an APPC program's RECEIVE_ALLOCATE verb or a CPI-C program's Accept_Conversation (CMACCP) call will wait for an incoming attach to arrive.

A value of zero indicates that there is no timeout."

::= { appcTpAdminEntry 7 }

appcTpAdminSyncLvl OBJECT-TYPE

SYNTAX INTEGER {
 none(1),
 confirm(2),
 noneAndConfirm(3),
 syncpoint(4),
 noneAndSyncpoint(5),

```

        confirmAndSyncpoint(6),
        all(7)
    }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Indicates the synchronization level or levels that the
    transaction program supports.  The levels are defined as
    follows:

        none      - allocation requests indicating a
                    synchronization level of none are allowed to
                    start the program.
        confirm   - allocation requests indicating a
                    synchronization level of confirm are allowed
                    to start the program.
        syncpoint - allocation requests indicating a
                    synchronization level of sync point are
                    allowed to start the program."

```

```
 ::= { appcTpAdminEntry 8 }
```

appcTpAdminInstLmt OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum number of concurrent instances of this transaction program that will be supported for a local LU. A value of zero indicates that there is no limit."

```
 ::= { appcTpAdminEntry 9 }
```

appcTpAdminStatus OBJECT-TYPE

```
SYNTAX INTEGER {
    enabled(1),
    tempDisabled(2),
    permDisabled(3)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the status of the TP relative to starting execution when the local LU receives an allocation (ATTACH) request naming this program.

```

        enabled      - the local LU can start the program.
        tempDisabled - the local LU cannot start the

```


program. The local LU rejects the request with an indication that the TP is not available but retry is possible.

permDisabled - the local LU cannot start the program. The local LU rejects the request with an indication that the TP is not available and retry is not possible."

::= { appcTpAdminEntry 10 }

appcTpAdminLongRun OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates whether this is a long-running transaction program (i.e., one that stays around even after the conversation goes away)."

::= { appcTpAdminEntry 11 }

appcTpAdminConvType OBJECT-TYPE

SYNTAX INTEGER {
 basic(1),
 mapped(2),
 basicOrMapped(3)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the conversation type (basic or mapped) that will be used by the TP. This value is verified upon receipt of an incoming attach. The acceptable values are:

basic - Indicates that this transaction program supports basic conversations.

mapped - Indicates that this transaction program supports mapped conversations.

basicOrMapped - Indicates that this transaction program supports both basic and mapped conversations."

```
::= { appcTpAdminEntry 12 }
```

```
appcTpAdminConvDuplex OBJECT-TYPE
```

```
SYNTAX INTEGER {  
    half(1),  
    full(2),  
    halfOrFull(3)  
}
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

"Specifies the conversation duplex type (half or full) that will be used by the TP. This value is verified upon receipt of an incoming attach. The acceptable values are:

- half - Indicates that this transaction program supports half duplex conversations.
- full - Indicates that this transaction program supports full duplex conversations.
- halfOrFull - Indicates that this transaction program supports either half or full duplex conversations."

```
::= { appcTpAdminEntry 13 }
```

```
appcTpAdminConvSecReq OBJECT-TYPE
```

```
SYNTAX INTEGER {  
    no(1),  
    yes(2)  
}
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

"Indicates whether conversation-level security information is required on incoming attaches designating this TP name. Conversation-level security verification is always performed on those requests that include security information.

- yes - Indicates that conversation-level security information is required. ATTACHs designating the transaction program must carry a user_id and either a password or an already verified indicator.
- no - Indicates that no security information is required. ATTACHs designating the transaction

program can omit or include security information."

::= { appcTpAdminEntry 14 }

appcTpAdminVerPip OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether the number of PIP (Program Initialization Parameters) subfields should be verified against the number expected (appcTpAdminPipSubNum)."

::= { appcTpAdminEntry 15 }

appcTpAdminPipSubNum OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the number of PIP subfields expected by the TP."

::= { appcTpAdminEntry 16 }

```
-- *****
-- APPC Active Session Table
-- -----
-- This table contains information about active APPC sessions.
-- *****
```

appcActSessTable OBJECT-TYPE

SYNTAX SEQUENCE OF AppcActSessEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of active APPC sessions. Two entries are present in the table when both session partners are local."

::= { appcSession 1 }

appcActSessEntry OBJECT-TYPE

SYNTAX AppcActSessEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entry of APPC Session Information Table. Indexed by LU pair and integer-valued session index."

```
INDEX { appcActSessLocLuName,
        appcActSessParLuName,
        appcActSessIndex }
```

```
::= { appcActSessTable 1 }
```

```
AppcActSessEntry ::= SEQUENCE {
    appcActSessLocLuName      DisplayString,
    appcActSessParLuName      DisplayString,
    appcActSessIndex          Integer32,
    appcActSessPcidCpName     DisplayString,
    appcActSessPcid           OCTET STRING,
    appcActSessPluIndicator    INTEGER,
    appcActSessModeName       DisplayString,
    appcActSessCosName        DisplayString,
    appcActSessTransPriority    INTEGER,
    appcActSessEnhanceSecSup    INTEGER,
    appcActSessSendPacingType  INTEGER,
    appcActSessSendRpc         Gauge32,
    appcActSessSendNxWndwSize  Gauge32,
    appcActSessRecvPacingType  INTEGER,
    appcActSessRecvRpc         Gauge32,
    appcActSessRecvNxWndwSize  Gauge32,
    appcActSessRscv           OCTET STRING,
    appcActSessInUse          INTEGER,
    appcActSessMaxSndRuSize    INTEGER,
    appcActSessMaxRcvRuSize    INTEGER,
    appcActSessSndPacingSize   INTEGER,
    appcActSessRcvPacingSize   INTEGER,
    appcActSessOperState       INTEGER,
    appcActSessUpTime          TimeTicks,
    appcActSessRtpNceId        OCTET STRING,
    appcActSessRtpTcid         OCTET STRING,
    appcActSessLinkIndex       InstancePointer
}
```

appcActSessLocLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Specifies the name of the local LU for the session. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is

present.

If this object has the same value as appcLluOperName, then the two entries being indexed apply to the same resource (specifically, to the same local LU)."

::= { appcActSessEntry 1 }

appcActSessParLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Specifies the name of the partner LU for the session. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

If this object has the same value as appcLuPairOperParLuName, then the two entries being indexed apply to the same resource (specifically, to the same partner LU)."

::= { appcActSessEntry 2 }

appcActSessIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This value identifies the index of the session, which is unique for this LU pair. It is recommended that an Agent not reuse the index of a deactivated session for a significant period of time (e.g., one week)."

::= { appcActSessEntry 3 }

appcActSessPcidCpName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0 | 3..17))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The network-qualified CP name of the node at which the session and PCID originated. For APPN and LEN nodes, this is either CP name of the APPN node at which the origin LU is located or the CP name of the NN serving the LEN node at which the origin LU is located. This field is from 3 to 17 characters in length, including a period (.) which separates the NetId from the NAU name. A null string indicates that the value is unknown."

```
::= { appcActSessEntry 4 }
```

appcActSessPcid OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0|8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The procedure correlation identifier (PCID) of a session. It is an 8-octet value assigned by the control point providing session services for the primary LU. A null string indicates that the value is unknown."

```
::= { appcActSessEntry 5 }
```

appcActSessPluIndicator OBJECT-TYPE

SYNTAX INTEGER {
 localLuIsPlu(1),
 partnerLuIsPlu(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates which LU is the PLU for this session. For independent LUs, the PLU (primary LU) is the one that initiated the session (sent BIND), while the SLU (secondary LU) is the one that accepted the session initiation (received BIND).

The 'local' LU is the one identified by appcLluOperName.

The 'partner' LU is the one identified by appcLuPairOperParLuName."

```
::= { appcActSessEntry 6 }
```

appcActSessModeName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The mode name used for this session."

```
::= { appcActSessEntry 7 }
```

appcActSessCosName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Class of Service (COS) name used for this session.
A null string indicates that the value is unknown."

::= { appcActSessEntry 8 }

appcActSessTransPriority OBJECT-TYPE

SYNTAX INTEGER {
 unknown(1),
 low(2),
 medium(3),
 high(4),
 network(5)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The transmission priority of this session.

1 = unknown priority

2 = low priority

3 = medium priority

4 = high priority

5 = network priority

"

::= { appcActSessEntry 9 }

appcActSessEnhanceSecSup OBJECT-TYPE

SYNTAX INTEGER {
 none(1),
 level1(2),
 level2(3)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Enhanced security supported. Indicates the level of enhanced
security support:

1 = none

2 = level 1

3 = level 2

"

::= { appcActSessEntry 10 }

appcActSessSendPacingType OBJECT-TYPE

SYNTAX INTEGER {
 none(1),

```

        fixed(2),
        adaptive(3)
    }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The type of pacing being used for sending data."

 ::= { appcActSessEntry 11 }

appcActSessSendRpc OBJECT-TYPE
    SYNTAX Gauge32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The send residual pace count.  This represents the number of
        MUs that can still be sent in the current session window."

    ::= { appcActSessEntry 12 }

appcActSessSendNxWndwSize OBJECT-TYPE
    SYNTAX Gauge32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The size of the next window which will be used to send data."

    ::= { appcActSessEntry 13 }

appcActSessRecvPacingType OBJECT-TYPE
    SYNTAX INTEGER {
        none(1),
        fixed(2),
        adaptive(3)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The type of pacing being used for receiving data."

    ::= { appcActSessEntry 14 }

appcActSessRecvRpc OBJECT-TYPE
    SYNTAX Gauge32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The receive residual pace count.  This represents the number
```


of MUs that can still be received in the current session window."

::= { appcActSessEntry 15 }

appcActSessRecvNxWndwSize OBJECT-TYPE

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The size of the next window which will be used to receive data."

::= { appcActSessEntry 16 }

appcActSessRscv OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The route selection control vector (RSCV CV2B) used for this session. It is present for APPN-level implementations. LEN-level implementations will return a null string. The internal format of this vector is described in SNA Formats. This object contains an uninterpreted copy of the control vector (including the length and key fields)."

::= { appcActSessEntry 17 }

appcActSessInUse OBJECT-TYPE

SYNTAX INTEGER {
 no(1),
 yes(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether the session is currently in use (i.e., in bracket with a conversation)."

::= { appcActSessEntry 18 }

appcActSessMaxSndRuSize OBJECT-TYPE

SYNTAX INTEGER (1..8192)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum RU size used on this session for sending RUs."

```
::= { appcActSessEntry 19 }
```

appcActSessMaxRcvRuSize OBJECT-TYPE

SYNTAX INTEGER (1..8192)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum RU size used on this session for receiving RUs."

```
::= { appcActSessEntry 20 }
```

appcActSessSndPacingSize OBJECT-TYPE

SYNTAX INTEGER (1..63)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The size of the send pacing window on this session."

```
::= { appcActSessEntry 21 }
```

appcActSessRcvPacingSize OBJECT-TYPE

SYNTAX INTEGER (1..63)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The size of the receive pacing window on this session."

```
::= { appcActSessEntry 22 }
```

appcActSessOperState OBJECT-TYPE

SYNTAX INTEGER {
 unbound (1),
 pendingBind (2),
 bound (3),
 pendingUnbind (4)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The value indicates the current operational state of the session.

'unbound (1)' - session has been unbound;
in this state it will be removed from the
session table by the Agent.

'pendingBind (2)' - this state has different
meanings for dependent and independent LUs;

for dependent LU - waiting for BIND from the host, for independent LU - waiting for BIND response. When a session enters this state, the corresponding entry in the session table is created by the Agent.

'bound (3)' - session has been successfully bound.

'pendingUnbind (4)' - session enters this state when an UNBIND is sent and before the rsp(UNBIND) is received.

Session deactivation:

If a session is in the operational state 'bound (3)' then setting the value of this object to 'unbound (1)' will initiate the session shutdown.

If a session is in the operational state 'pendingBind (2)' then setting the value of this object to 'unbound (1)' will initiate the session shutdown.

If a session is in the operational state 'pendingUnbind (4)' for an abnormally long period of time (e.g., three minutes) then setting the value of this object to 'unbound (1)' will change the session operational state to 'unbound (1)'. "

::= { appcActSessEntry 23 }

appcActSessUpTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The length of time the session has been active, measured in hundredths of a second."

::= { appcActSessEntry 24 }

appcActSessRtpNceId OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The local HPR Network Connection Endpoint of the session."

```
::= { appcActSessEntry 25 }
```

```
appcActSessRtpTcid OBJECT-TYPE
```

```
SYNTAX OCTET STRING (SIZE (0|8))
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The local RTP connection TCID of the session."
```

```
::= { appcActSessEntry 26 }
```

```
appcActSessLinkIndex OBJECT-TYPE
```

```
SYNTAX InstancePointer
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"This value identifies the link over which the session passes.
This value points to the row in the table containing
information on the link instance. (e.g., the sdlcLSAdminTable
of the SNA DLC MIB module). This object may be NULL if the
link is not specified or if a link is not applicable (as for
APPN-level nodes)."
```

```
::= { appcActSessEntry 27 }
```

```
-- *****
-- The following table contains session statistics for APPC sessions.
-- *****
```

```
appcSessStatsTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF AppcSessStatsEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"This table contains dynamic statistical information relating
to active APPC sessions. The entries in this table cannot be
created by a Management Station. Two entries are present in
the table when both session partners are local. This table is
populated only when the value of appcCntrlOperStat is
'active'."
```

```
::= { appcSession 2 }
```

```
appcSessStatsEntry OBJECT-TYPE
```

```
SYNTAX AppcSessStatsEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

DESCRIPTION

"Contains statistics information for an APPC session. Each entry is created by the Agent. Objects in this table have read-only access. Each session from appcActSessTable has one entry in this table."

INDEX { appcSessStatsLocLuName,
appcSessStatsParLuName,
appcSessStatsSessIndex }

::= { appcSessStatsTable 1 }

```
AppcSessStatsEntry ::= SEQUENCE {
    appcSessStatsLocLuName      DisplayString,
    appcSessStatsParLuName      DisplayString,
    appcSessStatsSessIndex      Integer32,
    appcSessStatsSentFmdBytes   Counter32,
    appcSessStatsSentNonFmdBytes Counter32,
    appcSessStatsRcvdFmdBytes   Counter32,
    appcSessStatsRcvdNonFmdBytes Counter32,
    appcSessStatsSentFmdRus     Counter32,
    appcSessStatsSentNonFmdRus  Counter32,
    appcSessStatsRcvdFmdRus     Counter32,
    appcSessStatsRcvdNonFmdRus  Counter32,
    appcSessStatsCtrUpTime      TimeTicks
}
```

appcSessStatsLocLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Specifies the name of the local LU for the session. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

If this object has the same value as appcLluOperName, then the two entries being indexed apply to the same resource (specifically, to the same local LU)."

::= { appcSessStatsEntry 1 }

appcSessStatsParLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Specifies the name of the partner LU for the session. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

If this object has the same value as appcLuPairOperParLuName, then the two entries being indexed apply to the same resource (specifically, to the same partner LU)."

::= { appcSessStatsEntry 2 }

appcSessStatsSessIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This value identifies the index of the session, which is unique for this LU pair. It is recommended that an Agent not reuse the index of a deactivated session for a significant period of time (e.g., one week).

If this object has the same value as appcActSessIndex for the same LU pair, then the two entries being indexed apply to the same resource (specifically, to the same session)."

::= { appcSessStatsEntry 3 }

appcSessStatsSentFmdBytes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of function management data (FMD) bytes sent by the local LU."

::= { appcSessStatsEntry 4 }

appcSessStatsSentNonFmdBytes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of non-function management data (non-FMD) bytes sent by the local LU."

::= { appcSessStatsEntry 5 }

appcSessStatsRcvdFmdBytes OBJECT-TYPE

```
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of function management data (FMD) bytes received by
    the local LU."
```

```
::= { appcSessStatsEntry 6 }
```

```
appcSessStatsRcvdNonFmdBytes OBJECT-TYPE
```

```
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of non-function management data (non-FMD) bytes
    received by the local LU."
```

```
::= { appcSessStatsEntry 7 }
```

```
appcSessStatsSentFmdRus OBJECT-TYPE
```

```
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of function management data (FMD) RUs sent by the
    local LU."
```

```
::= { appcSessStatsEntry 8 }
```

```
appcSessStatsSentNonFmdRus OBJECT-TYPE
```

```
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of non-function management data (non-FMD) RUs sent
    by the local LU."
```

```
::= { appcSessStatsEntry 9 }
```

```
appcSessStatsRcvdFmdRus OBJECT-TYPE
```

```
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of function management data (FMD) RUs received by
    the local LU."
```

```
::= { appcSessStatsEntry 10 }
```

appcSessStatsRcvdNonFmdRus OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of non-function management data (non-FMD) RUs received by the local LU."

```
::= { appcSessStatsEntry 11 }
```

appcSessStatsCtrUpTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The length of time the counters for this session have been active, measured in hundredths of a second."

```
::= { appcSessStatsEntry 12 }
```

```
-- *****
-- APPC Historical Session Table
-- -----
-- This table contains historical information about APPC sessions that
-- terminated abnormally. It is an implementation choice how long to
-- retain information on a given session.
-- *****
```

appcHistSessTable OBJECT-TYPE

SYNTAX SEQUENCE OF AppcHistSessEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of historical information about APPC sessions that terminated abnormally. Two entries may be present in the table when both session partners are local. It is an implementation choice how long to retain information about a given session."

```
::= { appcSession 3 }
```

appcHistSessEntry OBJECT-TYPE

SYNTAX AppcHistSessEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entry of APPC Session History Table. This table is indexed by an integer which is continuously incremented until it eventually wraps."

INDEX

```
{ appcHistSessIndex }
```

```
::= { appcHistSessTable 1 }
```

```
AppcHistSessEntry ::= SEQUENCE {
    appcHistSessIndex      INTEGER,
    appcHistSessTime       DateAndTime,
    appcHistSessType       INTEGER,
    appcHistSessLocLuName  DisplayString,
    appcHistSessParLuName  DisplayString,
    appcHistSessModeName   DisplayString,
    appcHistSessUnbindType OCTET STRING,
    appcHistSessSenseData  SnaSenseData,
    appcHistSessComponentId DisplayString,
    appcHistSessDetectModule DisplayString
}
```

appcHistSessIndex OBJECT-TYPE

```
SYNTAX INTEGER (0..2147483647)
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

DESCRIPTION

"Table index. The value of the index begins at zero and is incremented up to a maximum value of 2**31-1 (2,147,483,647) before wrapping."

```
::= { appcHistSessEntry 1 }
```

appcHistSessTime OBJECT-TYPE

```
SYNTAX DateAndTime
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

DESCRIPTION

"The time at which the session was either terminated or failed to be established."

```
::= { appcHistSessEntry 2 }
```

appcHistSessType OBJECT-TYPE

```
SYNTAX INTEGER {
    recvNegBindRsp(1),
    sendNegBindRsp(2),
    sessActRejected(3),
    unbindSent(4),
    unbindReceived(5)
}
```

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"Indicates the type of event that caused the entry to be made:

recvNegBindRsp - Received a negative bind response from
 the partner LU.
 sendNegBindRsp - Sent a negative bind response to the
 partner LU.
 sessActRejected - Session activation rejected by the
 partner LU.
 unbindSent - Unbind sent to the partner LU.
 unbindReceived - Unbind received from the partner LU.

These event types correspond to the five SNA/MS Alerts
 LU62001 through LU62005, documented in the SNA Management
 Services Reference."

::= { appcHistSessEntry 3 }

appcHistSessLocLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The network-qualified local LU name. This field is from 3 to
 17 characters in length, including a period (.) which separates
 the NetId from the NAU name if the NetId is present."

::= { appcHistSessEntry 4 }

appcHistSessParLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (3..17))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The network-qualified partner LU name. This field is from 3
 to 17 characters in length, including a period (.) which
 separates the NetId from the NAU name if the NetId is present."

::= { appcHistSessEntry 5 }

appcHistSessModeName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The mode name of the session."

```
::= { appcHistSessEntry 6 }
```

```
appcHistSessUnbindType OBJECT-TYPE
```

```
SYNTAX OCTET STRING (SIZE (1))
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The type of unbind which terminated the session. This
value is consists of one (1) octet; and its meaning
is defined in SNA Formats."
```

```
::= { appcHistSessEntry 7 }
```

```
appcHistSessSenseData OBJECT-TYPE
```

```
SYNTAX SnaSenseData
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The sense data associated with the termination of the
session, taken from the negative BIND response or UNBIND
request."
```

```
::= { appcHistSessEntry 8 }
```

```
appcHistSessComponentId OBJECT-TYPE
```

```
SYNTAX DisplayString (SIZE (0..32))
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The implementation-specific name of the component which
detected the problem."
```

```
::= { appcHistSessEntry 9 }
```

```
appcHistSessDetectModule OBJECT-TYPE
```

```
SYNTAX DisplayString (SIZE (0..32))
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The implementation-specific name of the module which
detected the problem."
```

```
::= { appcHistSessEntry 10 }
```

```
-- *****
-- APPC Session RTP Connection Table
-- -----
```

```
-- This table contains information on APPC sessions that are being
-- transported on RTP connections by High Performance Routing (HPR).
-- *****
appcSessRtpTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AppcSessRtpEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A table indicating how many APPC sessions terminating in this
        node are transported by each RTP connection."

    ::= { appcSession 4 }

appcSessRtpEntry OBJECT-TYPE
    SYNTAX AppcSessRtpEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Entry of APPC session RTP table."

    INDEX { appcSessRtpNceId,
            appcSessRtpTcid }

    ::= { appcSessRtpTable 1 }

AppcSessRtpEntry ::= SEQUENCE {
    appcSessRtpNceId          OCTET STRING,
    appcSessRtpTcid           OCTET STRING,
    appcSessRtpSessions       Gauge32
    }

appcSessRtpNceId OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE (1..8))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The local Network Connection Endpoint of the RTP connection."

    ::= { appcSessRtpEntry 1 }

appcSessRtpTcid OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE (8))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The local TCID of the RTP connection."

    ::= { appcSessRtpEntry 2 }
```

appcSessRtpSessions OBJECT-TYPE

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of APPC sessions terminating in this node that are using this RTP connection."

```
::= { appcSessRtpEntry 3 }
```

```
-- *****
--   APPC Active Conversation Table
--   This table contains information about active APPC conversations.
-- *****
```

appcActiveConvTable OBJECT-TYPE

SYNTAX SEQUENCE OF AppcActiveConvEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of information about active APPC Conversations. In this context 'active' means that a conversation is currently associated with a particular session. Two entries are present in the table when both LUs for the session are local."

```
::= { appcConversation 1 }
```

appcActiveConvEntry OBJECT-TYPE

SYNTAX AppcActiveConvEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entry representing one active APPC Conversation."

```
INDEX { appcActiveConvLocLuName,
        appcActiveConvParLuName,
        appcActiveConvSessIndex }
```

```
::= { appcActiveConvTable 1}
```

```
AppcActiveConvEntry ::= SEQUENCE {
```

```
    appcActiveConvLocLuName
```

```
    DisplayString,
```

```
    appcActiveConvParLuName
```

```
    DisplayString,
```

```
    appcActiveConvSessIndex
```

```
    Integer32,
```

```
    appcActiveConvId
```

```
    OCTET STRING,
```

```
    appcActiveConvState
```

```
    INTEGER,
```

```
    appcActiveConvType
```

```
    INTEGER,
```

appcActiveConvCorrelator	OCTET STRING,
appcActiveConvSyncLvl	INTEGER,
appcActiveConvSource	INTEGER,
appcActiveConvDuplex	INTEGER,
appcActiveConvUpTime	TimeTicks,
appcActiveConvSendBytes	Counter32,
appcActiveConvRcvBytes	Counter32,
appcActiveConvUserid	DisplayString,
appcActiveConvPcidNauName	DisplayString,
appcActiveConvPcid	OCTET STRING,
appcActiveConvModeName	DisplayString,
appcActiveConvLuwIdName	DisplayString,
appcActiveConvLuwIdInstance	OCTET STRING,
appcActiveConvLuwIdSequence	OCTET STRING,
appcActiveConvTpName	DisplayString
}	

appcActiveConvLocLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The SNA name of the local LU for the conversation. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

If this object has the same value as appcLluOperName, then the two entries being indexed apply to the same resource (specifically, to the same local LU)."

::= { appcActiveConvEntry 1 }

appcActiveConvParLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The SNA name of the partner LU for the conversation. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

If this object has the same value as appcLuPairOperParLuName, then the two entries being indexed apply to the same resource (specifically, to the same partner LU)."

::= { appcActiveConvEntry 2 }

appcActiveConvSessIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Index of entry in appcActSessTable that is associated with this conversation. If this object has the same value as appcActSessIndex for the same LU pair, then the two entries being indexed apply to the same resource (specifically, to the same session)."

::= { appcActiveConvEntry 3 }

appcActiveConvId OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (4))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The 4-byte ID of the conversation."

::= { appcActiveConvEntry 4 }

appcActiveConvState OBJECT-TYPE

SYNTAX INTEGER {

reset(1),

send(2),

receive(3),

confirm(4),

confirmSend(5),

confirmDealloc(6),

pendingDeallocate(7),

pendingPost(8),

sendReceive(9),

sendOnly(10),

receiveOnly(11),

deferReceive(12),

deferDeallocate(13),

syncpoint(14),

syncpointSend(15),

syncpointDeallocate(16),

backoutRequired(17)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the state of the conversation at the instant when the information was retrieved. The values are:

reset
The conversation is reset (i.e., deallocated).

send
The conversation can send data. This value also is returned if the conversation is in Send-Pending state.

receive
The conversation can receive data.

confirm
The conversation has received a confirm indicator. It can issue an [MC_]CONFIRMED or [MC_]SEND_ERROR verb to change state. It will continue in Receive state if an [MC_]CONFIRMED verb is issued.

confirmSend
The conversation is in Confirm state and changes to Send state when an [MC_]CONFIRMED verb is issued.

confirmDealloc
The conversation is in Confirm state and becomes inactive when an [MC_]CONFIRMED verb is issued.

pendingDeallocate
The conversation is in Pending-Deallocate state while it waits for (MC_)DEALLOCATE TYPE (sync_level) to complete.

pendingPost
The conversation is in Pending-Post state while it waits for the [MC_]RECEIVE_AND_POST verb to complete its receive function.

sendReceive
The full-duplex conversation can send or receive data.

sendOnly
The full-duplex conversation can send data, but it does not have permission to receive data, because the partner TP has already deallocated its side of the conversation.

receiveOnly
The full-duplex conversation can receive data, but it does not have permission to send data, because it has already deallocated its side of the conversation.

deferReceive
Waiting for a successful SYNCPT verb operation to go into the receive state.

deferDeallocate
Waiting for a successful SYNCPT verb operation to go into the reset state.

syncpoint
 Need to response to a SYNCPT verb issued. After successful operation, the next state will be receive.

syncpointSend
 Need to response to a SYNCPT verb issued. After successful operation, the next state will be send.

syncpointDeallocate
 Need to response to a SYNCPT verb issued. After successful operation, the next state will be reset.

backoutRequired
 TP must execute a BACKOUT verb to backout the transaction."

::= { appcActiveConvEntry 5 }

appcActiveConvType OBJECT-TYPE

SYNTAX INTEGER {
 basic(1),
 mapped(2)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the type of conversation. The values are:

basic

Indicates that this conversation supports basic verbs.

mapped

Indicates that this conversation supports mapped verbs."

::= { appcActiveConvEntry 6 }

appcActiveConvCorrelator OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This is an 8-byte identifier that the source LU assigns to the conversation; the source LU is the one that sent the allocation request. The conversation correlator is included on the allocation request. The conversation correlator uniquely

identifies a conversation, from among all conversations, between the local and partner LUs. It may be used, for example, during problem determination associated with a conversation. A length of 0 indicates that no conversation correlator is defined."

::= { appcActiveConvEntry 7 }

appcActiveConvSyncLvl OBJECT-TYPE

SYNTAX INTEGER {
 none(1),
 confirm(2),
 syncpt(3)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the highest sync level support for the conversation. The values are:

 none

Indicates that no sync-level processing can be performed on this conversation. The transaction program does not issue verbs or recognize any returned parameters relating to any sync-level function.

 confirm

Indicates that confirmation processing can be performed on this conversation. The transaction program can issue verbs and recognize returned parameters relating to confirmation.

 syncpt

Indicates that syncpt and confirmation processing can be performed on this conversation. The transaction program can issue verbs and recognize returned parameters relating to syncpt and confirmation."

::= { appcActiveConvEntry 8 }

appcActiveConvSource OBJECT-TYPE

SYNTAX INTEGER {
 localLu(1),
 partnerLu(2)
}

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"Indicates whether the local or partner LU is the source of the conversation, that is, which LU started the conversation by sending the allocation request.

localLu

The local LU is the source of the conversation, and the partner LU is the target of the conversation.

partnerLu

The partner LU is the source of the conversation, and the local LU is the target of the conversation."

::= { appcActiveConvEntry 9 }

appcActiveConvDuplex OBJECT-TYPE

SYNTAX INTEGER {
 half(1),
 full(2)
 }

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"Indicates the conversation duplex style in effect for the conversation.

half Indicates that information can be transferred in both directions, but only in one direction at a time.

full Indicates that information can be transferred in both directions at the same time."

::= { appcActiveConvEntry 10 }

appcActiveConvUpTime OBJECT-TYPE

SYNTAX TimeTicks
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The length of time since the conversation started, measured in hundredths of a second."

::= { appcActiveConvEntry 11 }

appcActiveConvSendBytes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the number of bytes that was sent on the conversation. The count includes all SNA RU bytes sent, including those in the FMH-5 (Attach), FMH-7 (Error Description), SIGNAL, LUSTAT, and SNA responses; it does not include SNA TH and RH bytes."

::= { appcActiveConvEntry 12 }

appcActiveConvRcvBytes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the number of bytes that was received on the conversation. The count includes all SNA RU bytes sent, including those in the FMH-5 (Attach), FMH-7 (Error Description), SIGNAL, LUSTAT, and SNA responses; it does not include SNA TH and RH bytes."

::= { appcActiveConvEntry 13 }

appcActiveConvUserid OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..10))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The user ID that the initiating program provided in the incoming attach."

::= { appcActiveConvEntry 14 }

appcActiveConvPcidNauName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0 | 3..17))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The network-qualified NAU name of the node at which the session and PCID originated. For APPN and LEN nodes, this is either CP name of the APPN node at which the origin LU is located or the CP name of the NN serving the LEN node at which the origin LU is located. This field is from 3 to 17 characters in length, including a period (.) which separates the

NetId from the NAU name. A null string indicates that the value is unknown."

::= { appcActiveConvEntry 15 }

appcActiveConvPcid OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0|8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The procedure correlation identifier (PCID) of the session.

It is an 8-octet value assigned by the control point providing session services for the primary LU. A null string indicates that the value is unknown."

::= { appcActiveConvEntry 16 }

appcActiveConvModeName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Mode Name used for this conversation.

This is a 1-8 character name."

::= { appcActiveConvEntry 17 }

appcActiveConvLuwIdName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The SNA name of the LU that initiated the logical unit of work that is associated with this active TP. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the LU name if the NetId is present."

::= { appcActiveConvEntry 18 }

appcActiveConvLuwIdInstance OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..6))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The instance identifier for the logical unit of work."

::= { appcActiveConvEntry 19 }

appcActiveConvLuwIdSequence OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..2))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The sequence identifier for the logical unit of work."

::= { appcActiveConvEntry 20 }

appcActiveConvTpName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..64))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The transaction program name which started this conversation. This name could either be from a FMH5 ATTACH for a remotely started conversation, otherwise it could be the name of the local TP if available.

When the TP name consists entirely of displayable EBCDIC code points, it is mapped directly to the equivalent ASCII display string. However, registered TP names always have a non-displayable EBCDIC code point (value less than or equal to x'3F') as the first character, so they cannot be directly mapped to an ASCII display string. These TP names are converted to a display string that is equivalent to a hexadecimal display of the EBCDIC code points. For example, the 2-byte TP name x'06F1' (CNOS) is converted to the 6-byte ASCII display string '06F1' (including the two single quotes).

This name is NULL if the conversation is started locally (i.e., not with a FMH5 ATTACH)."

::= { appcActiveConvEntry 21 }

```
-- *****
-- APPC Historical Conversation Table
-- This table contains historical information about APPC
-- conversations that ended abnormally. It is an implementation
-- choice how long to retain information on a given conversation.
-- *****
```

appcHistConvTable OBJECT-TYPE

SYNTAX SEQUENCE OF AppcHistConvEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of historical information about APPC Conversations that

ended in error. Possible categories of error conditions that could be saved in this table are:

- allocation errors,
- deallocate abend,
- program errors, and
- service errors."

```
::= { appcConversation 2 }
```

appcHistConvEntry OBJECT-TYPE

SYNTAX AppcHistConvEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entry representing one APPC Conversation."

INDEX

```
{ appcHistConvIndex }
```

```
::= { appcHistConvTable 1 }
```

AppcHistConvEntry ::= SEQUENCE {

appcHistConvIndex	Integer32,
appcHistConvEndTime	DateAndTime,
appcHistConvLocLuName	DisplayString,
appcHistConvParLuName	DisplayString,
appcHistConvTpName	DisplayString,
appcHistConvPcidNauName	DisplayString,
appcHistConvPcid	OCTET STRING,
appcHistConvSenseData	SnaSenseData,
appcHistConvLogData	OCTET STRING,
appcHistConvEndedBy	INTEGER
	}

appcHistConvIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Index for entry in Conversation table. This value identifies the unique index of the conversation. It is recommended that an Agent not reuse the index of a deactivated conversation for a significant period of time (e.g. one week)."

```
::= { appcHistConvEntry 1 }
```

appcHistConvEndTime OBJECT-TYPE

SYNTAX DateAndTime
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The time at which the conversation ended."

::= { appcHistConvEntry 2 }

appcHistConvLocLuName OBJECT-TYPE
 SYNTAX DisplayString (SIZE (1..17))
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The name of the local LU for this conversation. This field is
 from 1 to 17 characters in length, including a period (.) which
 separates the NetId from the NAU name if the NetId is present."

::= { appcHistConvEntry 3 }

appcHistConvParLuName OBJECT-TYPE
 SYNTAX DisplayString (SIZE (1..17))
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The SNA name of the partner LU for the conversation. This
 field is from 1 to 17 characters in length, including a period
 (.) which separates the NetId from the NAU name if the NetId is
 present."

::= { appcHistConvEntry 4 }

appcHistConvTpName OBJECT-TYPE
 SYNTAX DisplayString (SIZE (0..64))
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The transaction program name which started this conversation.
 This name could either be from a FMH5 ATTACH for a remotely
 started conversation, otherwise it could be the name of the local
 TP if available.

When the TP name consists entirely of displayable EBCDIC code points, it is mapped directly to the equivalent ASCII display string. However, registered TP names always have a non-displayable EBCDIC code point (value less than or equal to x'3F') as the first character, so they cannot be directly mapped to an ASCII display string. These TP names are converted to a display string that is equivalent to a

hexadecimal display of the EBCDIC code points. For example, the 2-byte TP name x'06F1' (CNOS) is converted to the 6-byte ASCII display string '06F1' (including the two single quotes).

This name is NULL if the conversation is started locally (i.e., not with a FMH5 ATTACH)."

::= { appcHistConvEntry 5 }

appcHistConvPcidNauName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0 | 3..17))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The network-qualified NAU name of the node at which the session and PCID originated. For APPN and LEN nodes, this is either CP name of the APPN node at which the origin LU is located or the CP name of the NN serving the LEN node at which the origin LU is located. This field is from 3 to 17 characters in length, including a period (.) which separates the NetId from the NAU name. A null string indicates that the value is unknown."

::= { appcHistConvEntry 6 }

appcHistConvPcid OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0|8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The procedure correlation identifier (PCID) of the session. It is an 8-octet value assigned by the control point providing session services for the primary LU. A null string indicates that the value is unknown."

::= { appcHistConvEntry 7 }

appcHistConvSenseData OBJECT-TYPE

SYNTAX SnaSenseData

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The sense data associated with the action that ended this conversation, e.g., FMH-7 or UNBIND."

::= { appcHistConvEntry 8 }

appcHistConvLogData OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..32))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The first 32 bytes of the data portion of the Log Data GDS Variable that is associated with the last FMH-7 that occurred on this conversation. If there was no Log Data GDS Variable associated with the FMH-7, this object is null.

This object reflects only the data portion of the Log Data GDS Variable (i.e. not the LL or GDS Id)."

```
::= { appcHistConvEntry 9 }
```

appcHistConvEndedBy OBJECT-TYPE

SYNTAX INTEGER {

localLu(1),

partnerLu(2)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates which LU ended the conversation."

```
::= { appcHistConvEntry 10 }
```

```
-- *****
-- APPC CPIC Admin Table
-- Objects in this table contain default or expected configuration
-- values for CPI-C side information.
-- *****
```

appcCpicAdminTable OBJECT-TYPE

SYNTAX SEQUENCE OF AppcCpicAdminEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"APPC CPI-C side information table."

```
::= { appcCPIC 1 }
```

appcCpicAdminEntry OBJECT-TYPE

SYNTAX AppcCpicAdminEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entry of APPC CPI-C side information Table."

```
INDEX { appcCpicAdminLocLuName,
        appcCpicAdminSymbDestName }
```

```
::= { appcCpicAdminTable 1 }
```

```
AppcCpicAdminEntry ::= SEQUENCE {
    appcCpicAdminLocLuName          DisplayString,
    appcCpicAdminSymbDestName       DisplayString,
    appcCpicAdminParLuAlias         DisplayString,
    appcCpicAdminParLuName          DisplayString,
    appcCpicAdminModeName           DisplayString,
    appcCpicAdminTpNameType         INTEGER,
    appcCpicAdminTpName             DisplayString,
    appcCpicAdminUserid             DisplayString,
    appcCpicAdminSecurity           INTEGER
}
```

```
appcCpicAdminLocLuName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..17))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
```

"The SNA name of the local LU to which this CPI-C side information definition applies. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

The reserved value '*ALL' indicates that the definition applies to all local LUs, and not just to a single local LU."

```
::= { appcCpicAdminEntry 1 }
```

```
appcCpicAdminSymbDestName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..8))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
```

"Specifies the symbolic destination name used by CPI-C applications to identify this definition."

```
::= { appcCpicAdminEntry 2 }
```

```
appcCpicAdminParLuAlias OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..8))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

"A local alias for the partner LU. If not known or

not applicable, this object contains a zero-length string."

::= { appcCpicAdminEntry 3 }

appcCpicAdminParLuName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The SNA name of the partner LU. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present."

::= { appcCpicAdminEntry 4 }

appcCpicAdminModeName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the mode name. A mode defines the characteristics for a group of sessions. The mode name can be blank (8 space characters)."

::= { appcCpicAdminEntry 5 }

appcCpicAdminTpNameType OBJECT-TYPE

SYNTAX INTEGER {
 normal(1),
 snaServiceTp(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether the TP name in appcCpicAdminTpName identifies a normal TP or an SNA service TP. In this context, a normal TP is one with a name consisting only of displayable characters, while an SNA service TP has a name containing octets that do not map to displayable characters."

::= { appcCpicAdminEntry 6 }

appcCpicAdminTpName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..64))

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"Specifies the name of the partner TP to be used when a CPI-C application initiates a conversation specifying this side information entry.

Display convention

When the TP name consists entirely of displayable EBCDIC code points, it is mapped directly to the equivalent ASCII display string. However, registered TP names always have a non-displayable EBCDIC code point (value less than or equal to x'3F') as the first character, so they cannot be directly mapped to an ASCII display string. These TP names are converted to a display string that is equivalent to a hexadecimal display of the EBCDIC code points. For example, the 2-byte TP name x'06F1' (CNOS) is converted to the 6-byte ASCII display string '06F1' (including the two single quotes)."

::= { appcCpicAdminEntry 7 }

appcCpicAdminUserid OBJECT-TYPE
 SYNTAX DisplayString (SIZE (0..10))
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The security userid, if any, associated with the side information definition."

::= { appcCpicAdminEntry 8 }

appcCpicAdminSecurity OBJECT-TYPE
 SYNTAX INTEGER {
 none(1),
 same(2),
 pgm(3),
 pgmStrong(4),
 distributed(5),
 mutual(6)
 }

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"Specifies the security information to be used for allocating the conversation.

none	- No security information.
same	- Use the security environment currently associated with this TP.
pgm	- Use the program-supplied user_id and password.
pgmStrong	- Use the program-supplied user_id and password. The local LU will insure that the password is not exposed in clear-text form on the physical network.
distributed	- Use the security environment and a distributed security system to generate the authentication information for this request. If distributed security tokens cannot be generated, then fail the conversation.
mutual	- Authenticate both the user to the destination system and the destination system to the user."

```
::= { appcCpicAdminEntry 9 }
```

```
-- *****
-- APPC CPIC Oper Table
-- Objects in this table contain current operational values, such
-- as state values or negotiated parameters, for CPI-C side
-- information.
-- *****
```

```
appcCpicOperTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AppcCpicOperEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "APPC CPI-C side information operational table."
```

```
::= { appcCPIC 2 }
```

```
appcCpicOperEntry OBJECT-TYPE
    SYNTAX AppcCpicOperEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Entry of APPC CPI-C side information Table."
```

```
INDEX { appcCpicOperLocLuName,
        appcCpicOperSymbDestName }
```

```
::= { appcCpicOperTable 1 }
```

```

AppcCpicOperEntry ::= SEQUENCE {
    appcCpicOperLocLuName      DisplayString,
    appcCpicOperSymbDestName   DisplayString,
    appcCpicOperParLuAlias     DisplayString,
    appcCpicOperParLuName      DisplayString,
    appcCpicOperModeName       DisplayString,
    appcCpicOperTpNameType     INTEGER,
    appcCpicOperTpName         DisplayString,
    appcCpicOperUserid         DisplayString,
    appcCpicOperSecurity       INTEGER
}

```

```

appcCpicOperLocLuName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..17))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

```

"The SNA name of the local LU to which this CPI-C side information definition applies. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present.

The reserved value '*ALL' indicates that the definition applies to all local LUs, and not just to a single local LU."

```
 ::= { appcCpicOperEntry 1 }
```

```

appcCpicOperSymbDestName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..8))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

```

"Specifies the symbolic destination name used by CPI-C applications to identify this definition."

```
 ::= { appcCpicOperEntry 2 }
```

```

appcCpicOperParLuAlias OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..8))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

```

"A local alias for the partner LU. If not known or not applicable, this object contains a zero-length string."

```
 ::= { appcCpicOperEntry 3 }
```

```

appcCpicOperParLuName OBJECT-TYPE

```

SYNTAX DisplayString (SIZE (1..17))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The SNA name of the partner LU. This field is from 1 to 17 characters in length, including a period (.) which separates the NetId from the NAU name if the NetId is present."

::= { appcCpicOperEntry 4 }

appcCpicOperModeName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..8))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the mode name. A mode defines the characteristics for a group of sessions. The mode name can be blank (8 space characters)."

::= { appcCpicOperEntry 5 }

appcCpicOperTpNameType OBJECT-TYPE

SYNTAX INTEGER {
 normal(1),
 snaServiceTp(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies whether the TP name in appcCpicOperTpName identifies a normal TP or an SNA service TP. In this context, a normal TP is one with a name consisting only of displayable characters, while an SNA service TP has a name containing octets that do not map to displayable characters."

::= { appcCpicOperEntry 6 }

appcCpicOperTpName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..64))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the name of the partner TP to be used when a CPI-C application initiates a conversation specifying this side information entry."

Display convention

When the TP name consists entirely of displayable EBCDIC code points, it is mapped directly to the equivalent ASCII display string. However, registered TP names always have a non-displayable EBCDIC code point (value less than or equal to x'3F') as the first character, so they cannot be directly mapped to an ASCII display string. These TP names are converted to a display string that is equivalent to a hexadecimal display of the EBCDIC code points. For example, the 2-byte TP name x'06F1' (CNOS) is converted to the 6-byte ASCII display string '06F1' (including the two single quotes)."

```
::= { appcCpicOperEntry 7 }
```

```
appcCpicOperUserid OBJECT-TYPE
```

```
SYNTAX DisplayString (SIZE (0..10))
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The security userid, if any, associated with the active side
information definition."
```

```
::= { appcCpicOperEntry 8 }
```

```
appcCpicOperSecurity OBJECT-TYPE
```

```
SYNTAX INTEGER {
```

```
    none(1),
```

```
    same(2),
```

```
    pgm(3),
```

```
    pgmStrong(4),
```

```
    distributed(5),
```

```
    mutual(6)
```

```
}
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Specifies the security information to be used for allocating
the conversation.
```

```
    none          - No security information.
```

```
    same          - Use the security environment currently
                    associated with this TP.
```

```
    pgm           - Use the program-supplied user_id and password.
```

```
    pgmStrong     - Use the program-supplied user_id and password.
                    The local LU will insure that the password is
                    not exposed in clear-text form on the physical
```

```

        network.
distributed - Use the security environment and a distributed
              security system to generate the authentication
              information for this request.  If distributed
              security tokens cannot be generated, then fail
              the conversation.
mutual       - Authenticate both the user to the destination
              system and the destination system to the user."

 ::= { appcCpicOperEntry 9 }

-- *****
-- Conformance information
-- *****

appcConformance      OBJECT IDENTIFIER ::= {appcMIB 2 }

appcCompliances      OBJECT IDENTIFIER ::= {appcConformance 1 }
appcGroups           OBJECT IDENTIFIER ::= {appcConformance 2 }

-- Compliance statements
appcCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for the SNMPv2 entities which
        implement the APPC MIB."

    MODULE -- this module

-- Unconditionally mandatory groups
MANDATORY-GROUPS {
    appcGlobalConfGroup,
    appcLluConfGroup,
    appcParLuConfGroup,
    appcModeConfGroup,
    appcTpConfGroup,
    appcSessionConfGroup
}

-- Conditionally mandatory groups
GROUP appcControlConfGroup
    DESCRIPTION
        "The appcControlConfGroup is mandatory only for those
        entities which implement activation and deactivation of
        specific controls such as statistics collecting and
        counting."

```

```

GROUP appcCnosConfGroup
  DESCRIPTION
    "The appcCnosConfGroup is mandatory only for those entities
    which implement CNOS. "

GROUP appcCpicConfGroup
  DESCRIPTION
    "The appcCpicConfGroup is mandatory only for those entities
    which implement CPI-C. "

GROUP appcConversationConfGroup
  DESCRIPTION
    "The appcConversationConfGroup is mandatory only for those
    entities which implement session endpoints for non-control
    APPC sessions."

-- MIN-ACCESS for objects
  OBJECT appcActSessOperState
  MIN-ACCESS read-only
  DESCRIPTION
    "An implementation is not required to support session
    deactivation via this object."

 ::= { appcCompliances 1 }

-- Units of conformance
appcGlobalConfGroup OBJECT-GROUP
  OBJECTS {
    appcUpTime,
    appcDefaultModeName,
    appcDefaultLuName,
    appcDefaultImplInbndPlu,
    appcDefaultMaxMcLlSndSize,
    appcDefaultFileSpec,
    appcDefaultTpOperation,
    appcDefaultTpConvSecRqd,
    appcLocalCpName,
    appcActiveSessions,
    appcActiveHprSessions
  }
  STATUS current
  DESCRIPTION
    "A collection of objects providing the instrumentation of APPC
    global information and defaults."

 ::= { appcGroups 1 }

```

appcLluConfGroup OBJECT-GROUP

```

OBJECTS {
    appcLluAdminDepType,
    appcLluAdminLocalAddress,
    appcLluAdminSessLimit,
    appcLluAdminBindRspMayQ,
    appcLluAdminCompression,
    appcLluAdminInBoundCompLevel,
    appcLluAdminOutBoundCompLevel,
    appcLluAdminCompRleBeforeLZ,
    appcLluAdminAlias,

    appcLluOperDepType,
    appcLluOperLocalAddress,
    appcLluOperSessLimit,
    appcLluOperBindRspMayQ,
    appcLluOperCompression,
    appcLluOperInBoundCompLevel,
    appcLluOperOutBoundCompLevel,
    appcLluOperCompRleBeforeLZ,
    appcLluOperAlias,
    appcLluOperActiveSessions
}

```

STATUS current

DESCRIPTION

"A collection of objects providing the instrumentation of APPC local LU6.2s."

```
::= { appcGroups 2 }
```

appcParLuConfGroup OBJECT-GROUP

```

OBJECTS {
    appcLuPairAdminParLuAlias,
    appcLuPairAdminSessLimit,
    appcLuPairAdminSessSec,
    appcLuPairAdminSecAccept,
    appcLuPairAdminLinkObjId,
    appcLuPairAdminParaSessSup,

    appcLuPairOperParLuAlias,
    appcLuPairOperSessLimit,
    appcLuPairOperSessSec,
    appcLuPairOperSecAccept,
    appcLuPairOperLinkObjId,
    appcLuPairOperParaSessSup,
    appcLuPairOperParaSessSupLS,
    appcLuPairOperState
}

```

STATUS current

DESCRIPTION

"A collection of objects providing the instrumentation of APPC partner LUs."

::= { appcGroups 3 }

appcModeConfGroup OBJECT-GROUP

```

OBJECTS {
    appcModeAdminCosName,
    appcModeAdminSessEndTpName,
    appcModeAdminMaxSessLimit,
    appcModeAdminMinCwinLimit,
    appcModeAdminMinClosLimit,
    appcModeAdminConWinAutoActLmt,
    appcModeAdminRecvPacWinSz,
    appcModeAdminSendPacWinSz,
    appcModeAdminPrefRecvRuSz,

    appcModeAdminPrefSendRuSz,
    appcModeAdminRecvRuSzUpBnd,
    appcModeAdminSendRuSzUpBnd,
    appcModeAdminRecvRuSzLoBnd,
    appcModeAdminSendRuSzLoBnd,
    appcModeAdminSingSessReinit,
    appcModeAdminCompression,
    appcModeAdminInBoundCompLevel,
    appcModeAdminOutBoundCompLevel,
    appcModeAdminCompRleBeforeLZ,
    appcModeAdminSyncLvl,
    appcModeAdminCrypto,

    appcModeOperCosName,
    appcModeOperSessEndTpName,
    appcModeOperSessLimit,
    appcModeOperMaxSessLimit,
    appcModeOperMinCwinLimit,
    appcModeOperMinClosLimit,
    appcModeOperConWinAutoActLmt,
    appcModeOperRecvPacWinSz,
    appcModeOperSendPacWinSz,
    appcModeOperPrefRecvRuSz,
    appcModeOperPrefSendRuSz,
    appcModeOperRecvRuSzUpBnd,
    appcModeOperSendRuSzUpBnd,
    appcModeOperRecvRuSzLoBnd,
    appcModeOperSendRuSzLoBnd,
    appcModeOperSingSessReinit,

```

```

    appcModeOperCompression,
    appcModeOperInBoundCompLevel,
    appcModeOperOutBoundCompLevel,
    appcModeOperCompRleBeforeLZ,
    appcModeOperSyncLvl,
    appcModeOperCrypto,
    appcModeOperSyncLvlLastStart,
    appcModeOperCryptoLastStart,
    appcModeOperCNOSNeg,
    appcModeOperActCwin,
    appcModeOperActClos,
    appcModeOperPndCwin,
    appcModeOperPndClos,
    appcModeOperPtmCwin,
    appcModeOperPtmClos,
    appcModeOperDrainSelf,
    appcModeOperDrainPart
  }

```

STATUS current

DESCRIPTION

"A collection of objects providing the instrumentation of APPC modes."

::= { appcGroups 4 }

appcTpConfGroup OBJECT-GROUP

```

  OBJECTS {
    appcTpAdminFileSpec,
    appcTpAdminStartParm,
    appcTpAdminTpOperation,
    appcTpAdminInAttachTimeout,
    appcTpAdminRcvAllocTimeout,
    appcTpAdminSyncLvl,
    appcTpAdminInstLmt,
    appcTpAdminStatus,
    appcTpAdminLongRun,
    appcTpAdminConvType,
    appcTpAdminConvDuplex,
    appcTpAdminConvSecReq,
    appcTpAdminVerPip,
    appcTpAdminPipSubNum
  }

```

STATUS current

DESCRIPTION

"A collection of objects providing the instrumentation of APPC Transaction Programs."

::= { appcGroups 5 }

appcSessionConfGroup OBJECT-GROUP

```
OBJECTS {
    appcActSessPcidCpName,
    appcActSessPcid,
    appcActSessPluIndicator,
    appcActSessModeName,
    appcActSessCosName,
    appcActSessTransPriority,
    appcActSessEnhanceSecSup,
    appcActSessSendPacingType,
    appcActSessSendRpc,
    appcActSessSendNxWndwSize,
    appcActSessRecvPacingType,
    appcActSessRecvRpc,
    appcActSessRecvNxWndwSize,
    appcActSessRscv,
    appcActSessInUse,
    appcActSessMaxSndRuSize,
    appcActSessMaxRcvRuSize,
    appcActSessSndPacingSize,
    appcActSessRcvPacingSize,
    appcActSessOperState,
    appcActSessUpTime,
    appcActSessRtpNceId,
    appcActSessRtpTcid,
    appcActSessLinkIndex,

    appcSessStatsSentFmdBytes,
    appcSessStatsSentNonFmdBytes,
    appcSessStatsRcvdFmdBytes,
    appcSessStatsRcvdNonFmdBytes,
    appcSessStatsSentFmdRus,
    appcSessStatsSentNonFmdRus,
    appcSessStatsRcvdFmdRus,
    appcSessStatsRcvdNonFmdRus,
    appcSessStatsCtrUpTime,

    appcHistSessTime,
    appcHistSessType,
    appcHistSessLocLuName,
    appcHistSessParLuName,
    appcHistSessModeName,
    appcHistSessUnbindType,
    appcHistSessSenseData,
    appcHistSessComponentId,
    appcHistSessDetectModule,

    appcSessRtpSessions
}
```

```

    }
    STATUS current
    DESCRIPTION
        "A collection of objects providing the instrumentation of APPC
        LU6.2 sessions."

    ::= { appcGroups 6 }

appcControlConfGroup OBJECT-GROUP
    OBJECTS {
        appcCntrlAdminStat,
        appcCntrlAdminRscv,
        appcCntrlAdminTrace,
        appcCntrlAdminTraceParm,
        appcCntrlOperStat,
        appcCntrlOperStatTime,
        appcCntrlOperRscv,
        appcCntrlOperRscvTime,
        appcCntrlOperTrace,
        appcCntrlOperTraceTime,
        appcCntrlOperTraceParm
    }
    STATUS current
    DESCRIPTION
        "A collection of objects providing the instrumentation of APPC
        control."

    ::= { appcGroups 7 }

appcCnosConfGroup OBJECT-GROUP
    OBJECTS {
        appcCnosCommand,
        appcCnosMaxSessLimit,
        appcCnosMinCwinLimit,
        appcCnosMinClosLimit,
        appcCnosDrainSelf,
        appcCnosDrainPart,
        appcCnosResponsible,
        appcCnosForce,
        appcCnosTargetLocLuName,
        appcCnosTargetParLuName,
        appcCnosTargetModeName
    }
    STATUS current
    DESCRIPTION
        "A collection of objects providing the instrumentation of APPC
        CNOS processing."

```



```
::= { appcGroups 8 }
```

```
appcCpicConfGroup OBJECT-GROUP
```

```
OBJECTS {
    appcCpicAdminParLuAlias,
    appcCpicAdminParLuName,
    appcCpicAdminModeName,
    appcCpicAdminTpNameType,
    appcCpicAdminTpName,
    appcCpicAdminUserid,
    appcCpicAdminSecurity,
    appcCpicOperParLuAlias,
    appcCpicOperParLuName,
    appcCpicOperModeName,
    appcCpicOperTpNameType,
    appcCpicOperTpName,
    appcCpicOperUserid,
    appcCpicOperSecurity
}
```

```
STATUS current
```

```
DESCRIPTION
```

```
"A collection of objects providing the instrumentation of APPC
CPI-C side information."
```

```
::= { appcGroups 9 }
```

```
appcConversationConfGroup OBJECT-GROUP
```

```
OBJECTS {
    appcActiveConvId,
    appcActiveConvState,
    appcActiveConvType,
    appcActiveConvCorrelator,
    appcActiveConvSyncLvl,
    appcActiveConvSource,
    appcActiveConvDuplex,
    appcActiveConvUpTime,
    appcActiveConvSendBytes,
    appcActiveConvRcvBytes,
    appcActiveConvUserid,
    appcActiveConvPcidNauName,
    appcActiveConvPcid,
    appcActiveConvModeName,
    appcActiveConvLuwIdName,
    appcActiveConvLuwIdInstance,
    appcActiveConvLuwIdSequence,
    appcActiveConvTpName,

    appcHistConvEndTime,
```

```
        appcHistConvLocLuName,
        appcHistConvParLuName,
        appcHistConvTpName,
        appcHistConvPcidNauName,
        appcHistConvPcid,
        appcHistConvSenseData,
        appcHistConvLogData,
        appcHistConvEndedBy
    }
STATUS current
DESCRIPTION
    "A collection of objects providing the instrumentation of APPC
    conversations."

 ::= { appcGroups 10 }

-- end of conformance statement

END
```

5. Acknowledgments

This MIB module is the product of the SNA NAU MIB Working Group. Special thanks to Wayne Clark, Cisco Systems; Rich Daugherty, IBM Corporation; and Leo Temoshenko, IBM Corporation, for their contributions and review.

6. References

- [1] IBM, Systems Network Architecture Technical Overview, GC30-3073-4, January, 1994.
- [2] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, January 1996.
- [3] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.
- [4] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, January 1996.
- [5] IBM, Systems Network Architecture Transaction Programmer's Reference for LU Type 6.2, GC30-3084-05, June, 1993.
- [6] IBM, Common Programming Interface Communications Specification 2.0, SC31-6180-01, June, 1994.
- [7] Kielczewski, Z., Kostick D., and K. Shih, "Definition of Managed Objects for SNA NAUs using SMIV2", RFC 1666, Eicon Technology Corporation, Bell Communications Research, Novell, August 1994.

7. Security Considerations

Security issues are not discussed in this memo.

8. Authors' Addresses

Michael Allen
Wall Data Inc.
P.O.Box 1120, WA 98019, USA

Phone: +1 206 844 3505
EMail: mallen@hq.walldata.com

Bob Clouston
Cisco Systems
7025 Kit Creek Road
PO Box 14987
Research Triangle Park, NC 27709, USA

Phone: +1 919 472 2333
EMail: clouston@cisco.com

Zbigniew Kielczewski
Cisco Systems
7025 Kit Creek Road
PO Box 14987
Research Triangle Park, NC 27709, USA

Phone: +1 919 472 2326
EMail: zbig@cisco.com

William Kwan
Jupiter Technology Inc.
200 Prospect St.
Waltham, MA 02254

Phone: 1 617 894-9300 x423
EMail: billk@jti.com

Bob Moore
IBM Corporation
800 Park Offices Drive
E87/664
P.O. Box 12195
Research Triangle Park, NC 27709, USA

Phone: 1 919 254 4436
EMail: remoore@ralvm6.vnet.ibm.com

